

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
імені ІГОРЯ СІКОРСЬКОГО»**

Факультет електроенерготехніки та автоматики

Кафедра автоматизації електромеханічних систем та електроприводу

«На правах рукопису»  
УДК 62-52

«До захисту допущено»

Завідувач кафедри

\_\_\_\_\_  
(підпис) Сергій ПЕРЕСАДА

“ \_\_\_\_ ” \_\_\_\_\_ 20\_\_ р.

**Магістерська дисертація**

**на здобуття ступеня магістра**

**за освітньо-професійною програмою «Електромеханічні системи автомати-  
зації, електропривод та електромобільність»**

**зі спеціальності 141 «Електроенергетика, електротехніка  
та електромеханіка»**

**на тему:**

\_\_\_\_\_  
«Електромеханічна система багатоярусної паркінгової системи автомобі-  
лів»

Виконав: студент VI (II) курсу, групи ЕП-91мп

\_\_\_\_\_  
Жицький Богдан Олексійович  
(прізвище, ім'я, по-батькові)



Науковий керівник доц., д.т.н. Шаповал Іван Андрійович  
(посада, науковий ступінь, вчене звання, прізвище, ім'я, по-батькові)

  
(підпис)

Консультант \_\_\_\_\_  
(назва розділу) (посада, науковий ступінь, вчене звання, прізвище, ім'я, по-батькові) (підпис)

Рецензент доц, к.т.н. Реуцький Микола Олександрович  
(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище, ім'я, по-батькові)

  
(підпис)

Засвідчую, що у цій магістерській дисе-  
ртації немає запозичень з праць інших  
авторів без відповідних посилань.

Студент (-ка) \_\_\_\_\_  
(підпис)

Київ – 2020 р.

**Національний технічний університет України  
«Київський політехнічний інститут імені Ігоря Сікорського»**

Факультет електроенерготехніки та автоматики

Кафедра автоматизації електромеханічних систем та електроприводу

Рівень вищої освіти – другий (магістерський)

Спеціальність – 141 «Електроенергетика, електротехніка та електромеханіка»

Освітньо-професійна програма – «Електромеханічні системи автоматизації, електропривод та електромобільність»

ЗАТВЕРДЖУЮ

Завідувач кафедри

\_\_\_\_\_ Сергій ПЕРЕСАДА  
(підпис)

«\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**

**на магістерську дисертацію студенту**

Жицькому Богдану Олексійовичу

(прізвище, ім'я, по батькові)

1. Тема дисертації: «Електромеханічна система багатоярусної паркінгової системи автомобілів» \_\_\_\_\_,

науковий керівник дисертації: доц., д.т.н. Шаповал Іван Андрійович \_\_\_\_\_,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «\_\_\_» \_\_\_\_\_ 2020 р. № \_\_\_\_\_

2. Строк подання студентом дисертації: 11 грудня 2020 р.

3. Об'єкт дослідження: Паркінгові установки, їх види та способи автоматизації, оптимізація процесів в роботі механізмів, введення сучасних способів доступу до системи.

4. Вихідні дані: Базові типи автоматизованих паркінгових систем, що набувають поширення закордоном

5. Перелік завдань, які потрібно розробити: 1. Аналітичний огляд в області електромеханічних систем паркінгових систем автомобілів; 2. Визначення параметрів електромеханічної системи багатоярусної паркінгової системи автомобілів; 3. Вибір електродвигунів та електронних компонентів. Розробка (опис) схеми; 4. Вибір і розробка апаратного забезпечення; 5. Розробка програмного

забезпечення; 6. Дослідження роботи електромеханічної системи багатоярусної паркінгової системи автомобілів на експериментальному стенді; 7. Стартап проєкт.

6. Перелік графічного (ілюстративного) матеріалу: Креслення загального виду установки; креслення виконавчих механізмів установки; схема електрична принципова; схема електрична з'єднань; пульт керування та алгоритми роботи установки; осцилограми роботи електромеханічної системи установки.

7. Орієнтовний перелік публікацій: Пушкар М.В. Концепція створення лабораторного стенду для дослідження роботи системи багатоярусної парковки на базі контролера Siemens Simatic / Б. О. Жицький, М. В. Пушкар. // Міжнародний науково-технічний журнал молодих учених, аспірантів і студентів "СУЧАСНІ ПРОБЛЕМИ ЕЛЕКТРОЕНЕРГОТЕХНІКИ ТА АВТОМАТИКИ". – 2019. – С. 403–408.; Жицький Б. О. Макет транспортно-складської системи для дослідження програм автоматизації / Б. О. Жицький, С. В. Король. // Всеукраїнська науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи». – 2020.

#### 8. Консультанти розділів дисертації\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

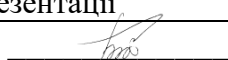
\* Якщо визначені консультанти. Консультантом не може бути зазначено наукового керівника магістерської дисертації.

#### 9. Дата видачі завдання: 01 вересня 2020 р.

##### Календарний план

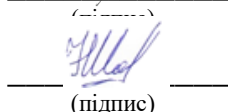
№ з/п	Назва етапів виконання магістерської дисертації	Строк виконання етапів магістерської дисертації	Примітка
1	Аналітичний огляд в області електромеханічних систем паркінгових систем автомобілів	11.09.2020	
2	Визначення параметрів електромеханічної системи багатоярусної паркінгової системи автомобілів	17.09.2020	
3	Вибір електродвигунів та електронних компонентів. Розробка (опис) схеми	02.10.2020	
4	Вибір і розробка апаратного забезпечення	16.10.2020	
5	Розробка програмного забезпечення	30.10.2020	
6	Дослідження роботи електромеханічної системи багатоярусної паркінгової системи автомобілів на експериментальному стенді	20.11.2020	
7	Стартап проєкт	27.11.2020	
	Оформлення пояснювальної записки та графічної частини. Підготовка презентації	04.12.2020	

Студент



Б.О. Жицький

Науковий керівник дисертації

  
(підпис)

І.А. Шаповал

## РЕФЕРАТ



Магістерська дисертація містить: сторінок – 134, рисунків – 62, таблиць – 7 та 49 посилань.

В магістерській дисертації проведено дослідження сучасних систем автоматизації паркінгових систем, їх класифікація, призначення та побудови. Визначено переваги та недоліки найбільш популярних із них, обрано одну систему для більш поглибленого дослідження.

Було створено об'ємну модель прототипу дослідження для подальшого виготовлення робочого макету. Для виготовлення певних деталей системи використано метод об'ємного друку пластиком.

Проведено дослідження режимів роботи двигунів механізму підйому та відвантаження, знято осцилограми. Виконавши дослідження, було встановлено, що використання у алгоритмах керування методики штучного обмеження «мертвих зон» регулювання значно полегшує пуск двигунів та покращує динаміку системи загалом.

ДВИГУН ПОСТІЙНОГО СТРУМУ, ШИРОТНО-ІМПУЛЬСНА МОДУЛЯЦІЯ, ПАРКІНГ, ДОСЛІДЖЕННЯ АЛГОРИТМУ КЕРУВАННЯ, ANDROID, ПРОГРАМУВАННЯ, МЕРТВА ЗОНА, ЄМНІСНИЙ ФІЛЬТР

					141.62023.029.БР						
		№ докум.	Підпис								
Розроб.	Жицький Б.О.				«Електромеханічна система багато- ярусної паркінгової системи автомо- білів»	Літ.	Арк.	Аркушів			
Перевір.	Шаповал І.А.							4	134		
						НТУУ «КПІ», ФЕА Кафедра АЕМС-ЕП Гр. ЕП-91мп					
Н. Контр.											
Затверд.											

## SUMMARY



Master's diploma work contains pages - 134, figures - 62, tables - 7 and 49 references.

In this master's dissertation the research of modern systems of automation of parking systems, their classification, purpose and structure are carried out. The advantages and disadvantages of the most popular of them are identified, one system is chosen for more in-depth research.

A three-dimensional model of the research prototype was created for further production of the current model. The method of volume printing with plastic is used to make certain parts of the system.

A study of the modes of operation of the engines of the lifting and loading mechanism, oscillograms were taken. After conducting research, it was found that the use in control algorithms of the technique of artificial restriction of "dead zones" of regulation significantly facilitates the start of engines and improves the dynamics of the system as a whole.

DC MOTOR, PULSE WIDTH MODULATION, PARKING, CONTROL ALGORITHM RESEARCH, ANDROID, PROGRAMMING, DEAD ZONE, CAPACITY FILTER

					141.62023.029.BW			
		№ of doc.	Sign.					
Devel.		B. Zhitskiy						
Checked		I. Shapoval						
N. Contr.								
Approved.								
						L.	Page	Pages
							5	134
						NTUU «KPI», FEA Department AEMS-ED gr. EP-9Imp		

## ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ .....	11
ВСТУП .....	12
РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД В ОБЛАСТІ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ ПАРКІНГОВИХ СИСТЕМ АВТОМОБІЛІВ .....	16
1.1 Типові технічні рішення .....	19
1.1.1 Механізовані парковки баштового типу .....	19
1.1.2 Механізовані парковки, площинні, з проїздом по центру .....	22
1.1.3 Механізовані парковки, площинні, без проїзду по центру .....	23
1.1.4 Підводні паркінги .....	24
1.1.5 Паркувальна система BOX PARKING .....	25
1.1.6 Паркувальний підйомник МЕТРО 2Р .....	26
1.1.7 Паркувальна система ТРАНСЛО ПАРКІНГ .....	27
1.1.8 Паркувальна платформа ТТ/ТЛ .....	29
1.2 Опис електромеханічної системи багаторушної паркінгової системи автомобілів .....	30
1.2.1 Опорні стійки .....	30
1.2.2 Паркувальна палета .....	31
1.2.3 Система синхронізації .....	31
1.2.4 Електрогідравлічний блок .....	31
1.2.5 Шафа електрична .....	32
1.2.6 Панель управління .....	32

1.3 Формулювання вимог до систем автоматичного керування багатоярусної паркінгової системи автомобілів .....	33
Висновки до розділу .....	35
РОЗДІЛ 2. ВИЗНАЧЕННЯ ПАРАМЕТРІВ ЕЛЕКТРОМЕХАНІЧНОЇ СИСТЕМИ БАГАТОЯРУСНОЇ ПАРКІНГОВОЇ СИСТЕМИ АВТОМОБІЛІВ .....	36
2.1 Прототип конструкції проектного стенду .....	36
2.2 Огляд електротехнічної складової проектного стенду .....	42
2.2.1. Електропривідна частина .....	42
2.2.2 Контролер .....	44
2.2.3 Датчики .....	45
Висновки до розділу .....	46
РОЗДІЛ 3. ВИБІР ЕЛЕКТРОДВИГУНІВ ТА ЕЛЕКТРОННИХ КОМПОНЕНТІВ. РОЗРОБКА (ОПИС) СХЕМИ .....	47
3.1 Вибір та опис двигуна .....	47
3.2 Вибір драйверу для керування мотор-редуктором .....	48
3.3 Індуктивний датчик положення LJ8A3-1-Z-AX NPN NC .....	51
3.4 Модуль бездротового зв'язку Bluetooth HC05 .....	52
3.5 Модуль бездротового зв'язку Wi-Fi ESP-01 .....	54
3.6 RFID модуль RC522 з картою доступу для Arduino .....	56
3.7 Система живлення пристроїв .....	58
Висновки до розділу .....	61
РОЗДІЛ 4. ВИБІР І РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ .....	62
4.1 Використання ПЛК на базі Arduino в практиці .....	62
4.2 Входи і виходи .....	64

4.3 Алгоритм відпрацювання завдання проєкту .....	66
4.4 Блок-схема алгоритмів .....	68
4.5 Інтерфейс користувача .....	68
Висновки до розділу .....	71
<b>РОЗДІЛ 5. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ .....</b>	<b>72</b>
5.1 Управління ДПС за допомогою ARDUINO .....	72
5.1.1 Ініціалізація бібліотеки .....	72
5.1.2 Налаштування бібліотеки .....	73
5.1.3 Режим роботи двигунів .....	73
5.1.4 Управління швидкістю і напрямком .....	74
5.1.5 Роздільна здатність ШІМ .....	74
5.1.6 Плавне керування швидкістю .....	75
5.2 Програмування модуля Bluetooth HC05 .....	75
5.3 Програмування RFID модуля RC522 з картою доступу для Arduino .....	77
5.3.1 Бібліотека MFRC522 .....	79
5.3.2 Запис і читання даних користувача .....	81
5.4 Розробка Android – додатку в середовищі App Inventor 2 .....	83
Висновки до розділу .....	96
<b>РОЗДІЛ 6. ДОСЛІДЖЕННЯ РОБОТИ ЕЛЕКТРОМЕХАНІЧНОЇ СИСТЕМИ БАГАТОЯРУСНОЇ ПАРКІНГОВОЇ СИСТЕМИ АВТОМОБІЛІВ НА ЕКСПЕРИМЕНТАЛЬНОМУ СТЕНДІ .....</b>	<b>98</b>
6.1 Рівні сигналів, що надходять від датчиків .....	98
6.1.1 Індуктивний датчик перешкод .....	98
6.1.2 Оптичний (інфрачервоний) датчик перешкод .....	99



6.2 Дослідження роботи двигунів у випадку керування методом ШІМ у статичному режимі .....	99
6.2.1 ШІМ-сигнал від Arduino до драйверів L298N .....	100
6.2.2 ШІМ-сигнал від драйверів L298N до двигунів механізму відвантаження автомобіля .....	101
6.2.3 ШІМ сигнал від драйверів L298N до двигунів механізму відвантаження автомобіля після встановлення керамічного конденсатора ємністю 0.1 мкФ .....	102
6.2.4 ШІМ-сигнал від драйверів L298N до двигунів підйомного механізму .....	104
6.2.5 ШІМ сигнал від драйверів L298N до двигунів підйомного механізму автомобіля після встановлення керамічного конденсатора ємністю 0.1 мкФ .....	105
6.3 Регулювання ШІМ для досягнення плавного наближення до цільової позиції.....	106
6.4 Вплив зміни максимальної швидкості руху двигунів на точність позиціонування, аварійність тощо.....	107
6.5 Розгляд можливості позиціонування механізмів без використання датчиків (аварійний режим) .....	108
Висновки до розділу.....	109
РОЗДІЛ 7. СТАРТАП ПРОЄКТ .....	110
7.1 Мета проєкту.....	110
7.2 Актуальність проєкту .....	110
7.3 Характеристика концепції проєкту.....	113
7.4 Фінансування проєкту .....	116
Інвестиції.....	116

	10
Система оподаткування.....	117
7.5 Співробітники і персонал.....	118
7.6 Просування і збільшення продажів .....	119
7.7 Фінансові результати.....	120
7.8 Аналіз ризиків.....	120
Висновки до розділу.....	121
ЗАГАЛЬНІ ВИСНОВКИ.....	122
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	124
ДОДАТОК 1 .....	128
ДОДАТОК 2 .....	130
ДОДАТОК 3 .....	131
ДОДАТОК 4 .....	134

## **ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ**

ПЗ – програмне забезпечення;

ОС – операційна система;

СНГ – співдружність незалежних держав (ор. Содружество Независимых Государств);

АСП – автоматизована система паркінгу;

АСУ – автоматизована система управління;

SoC – система-на-чипі;

ККД – коефіцієнт корисної дії;

WiFi, GSM/GPRS, IR, Bluetooth, RFID – технології безпроводного зв'язку;

КД – кроковий двигун;

ШІМ – широтно-імпульсна модуляція;

PLC / ПЛК – програмований логічний контролер;

ПК – персональний комп'ютер;

ДПС – двигун постійного струму;

СКД – система контролю доступу;

ДБН – державні будівельні норми;

ПДФО – податок на прибуток фізичних осіб;

ЗП – заробітна плата;

## ВСТУП

**Актуальність роботи.** Паркувальні комплекси, оснащені спеціалізованим обладнанням, увійшли в повсякденне життя європейців ще на початку 1980-х рр. До середини 1990-х рр., коли такі автоматизовані системи почали з'являтися і в країнах СНД, Європа встигла накопичити солідний досвід у цій галузі. В Україні ж необхідність в організованому паркуванні особистого автотранспорту зростала поступово і з кожним днем набувала все більшої актуальності.

Перші автоматизовані паркування в країнах СНД ставилися більше для додання значущості споруджуваним комплексам, а не як необхідний елемент інфраструктури. Ситуація кардинально змінилася, коли з'явилася проблема безпечного паркування особистого автотранспорту. Сьогодні наявність паркувальних місць у житлових комплексах, торгових і бізнес-центрах сприймається як само собою зрозуміле, а коли їх немає, це призводить до незручностей для автовласників.

Попит на паркувальні місця, що збільшується, призвів до потреби їхньої організації. Залишаючи машину на паркінгу, автовласник хоче зберегти не тільки її, але і свої нерви від простоювання в пробках, що утворюються в місцях проїзду через неструктурований рух. Організований паркінг може повністю задовольнити потреби автовласника. Впровадження автоматизації дає змогу зменшити час, що витрачається на в'їзд і виїзд, отже, уникнути затору в місцях проїзду, а оплата, виконана через автоматичний термінал, усуває конфліктні ситуації з операторами паркування.

Середньостатистичний паркінг має в собі в'їзд і виїзд на паркувальних стійках і оплату через автоматичний платіжний термінал. Відвідувач на такому паркінгу як би сам себе припарковує. Це швидко, зручно й легко. Втручання оператора в таких системах відбувається тільки в разі виникнення нештатних ситуацій.

Проектуючи автоматизовані системи, виробники прагнуть врахувати два нюанси – зручність для відвідувачів в експлуатації паркінгу і зручність для власників у його обслуговуванні та контролі. Усе це досягається завдяки чіткій організації взаємодії обладнання та програмного забезпечення (ПЗ).

Для власників паркінгів впровадження автоматизованих систем із картовою оплатою, чи за допомогою штрихкодів є вигідним. Вони дають змогу убезпечити власника від зловживання операторами паркування своїми повноваженнями. Крім отримання чистого прибутку, такі системи дають можливість стежити за всіма операціями, отримувати повну звітність за ними. Водночас така інформація може бути доступна через інтернет.

**Зв'язок роботи з науковими програмами, планами, темами.** Магістерську дисертацію виконано на кафедрі «Автоматизації електромеханічних систем, електропривод та електромобільність» Національного технічного університету України «Київський політехнічний інститут ім.Ігоря Сікорського» відповідно до плану підготовки магістрів.

**Мета магістерської дисертації.** Метою магістерської дисертації є проектування та виготовлення макету автоматизованої паркувальної системи, керованою за допомогою встановленого на Андроїд-пристрій додатку та RFID-мітки (для авторизації користувача). Макет має максимально детально візуалізувати реальний процес паркування персонального авто з урахуванням механічних процесів. Обов'язкова наявність захисту від не правильної постановки авто на платформу (палету), забезпечена плавність пересування платформи, жорсткість конструкції.

**Для досягнення поставленої мети необхідно вирішити такі завдання:**

- виконати аналітичний огляд літературних джерел та зробити опис технологічного процесу;
- виконати проектування 3D-моделі макету;
- виконати вибір силового обладнання з урахуванням навантаження;

- виконати вибір елементів керування, датчиків, механізмів тощо;
- виконати візуалізацію процесу постановки авто на паркомісце;
- написати Андроїд-додаток, який забезпечить зв'язок пристроєм керування (планшет, телефон) за допомогою однієї з безпроводних технологій;
- реалізувати метод авторизації користувача за допомогою RFID-картки чи брелока.

**Об'єкт дослідження** – процеси керування електромеханічною системою багатоярусної паркінгової системи автомобілів.

**Предмет дослідження** – системи контролю та керування доступом систем автоматизації технологічних процесів паркувальних систем.

**Методи дослідження.** У роботі використані фундаментальні положення теорії електропривода, теоретичної механіки, комп'ютерне моделювання.

**Наукова новизна одержаних результатів.** Розроблено процедуру проєктування основних елементів автоматизованої багатоярусної паркінгової системи автомобілів.

Отримано об'ємну модель паркувальної системи баштового типу, що дало змогу виконати дослідження системи точного позиціонування із використанням лічильника на основі безконтактного давача.

#### **Практичне значення одержаних результатів.**

Розроблена замкнена система керування на базі двигунів постійного струму, безконтактних давачів та технологій дає змогу проводити дослідження технологічного процесу паркування без участі персоналу.

Розроблені процедури можуть бути використані під час розроблення та проєктування електромеханічних систем автоматизованих багатоярусних паркінгів.

**Публікації.** Пушкар М.В. Концепція створення лабораторного стенду для дослідження роботи системи багатоярусної парковки на базі контролера Siemens Simatic / Б. О. Жицький, М. В. Пушкар. // Міжнародний науково-технічний журнал молодих учених, аспірантів і студентів "СУЧАСНІ

ПРОБЛЕМИ ЕЛЕКТРОЕНЕРГОТЕХНІКИ ТА АВТОМАТИКИ". – 2019. – С. 403–408.; Жицький Б. О. Макет транспортно-складської системи для дослідження програм автоматизації / Б. О. Жицький, С. В. Король. // Всеукраїнська науково-практична інтернет-конференція «Молодь в науці: дослідження, проблеми, перспективи». – 2020.

## **РОЗДІЛ 1. АНАЛІТИЧНИЙ ОГЛЯД В ОБЛАСТІ ЕЛЕКТРОМЕХАНІЧНИХ СИСТЕМ ПАРКІНГОВИХ СИСТЕМ АВТОМОБІЛІВ**

Автоматизовані паркувальні системи – окремі будівлі або вбудовані у будівлі конструкції для компактного зберігання автомобілів. Вони влаштовані за допомогою спеціальних підйомників, на які заїжджає автомобіль, після чого стоянкова платформа разом з автомобілем піднімається і рухається до місця паркування автомобіля самостійно. Завдяки зручній транспортування автомобіля, такі системи займають дуже мало місця, у порівнянні зі стандартними парковками, так як виключаються необхідність в спеціальних пандусах для заїзду на парковку та скорочується відстань, яку необхідно проїхати автомобілю, щоб під'їхати до місця стоянки і припаркуватися.

Переміщення людей на таких підйомних механізмах суворо заборонено. Рухомі платформи є відкритими і не обладнані стінами, поручнями або обмежувачами, що небезпечно для людини. Там, де безпосередньо зберігаються автомобілі, теж виключається перебування людей, так як автоматичні механізми, що переміщують автомобілі можуть зачепити і травмувати людину [1]..

Встановити автомобільний підйомник, призначений тільки для автомобіля, набагато вигідніше за ціною, ніж повноцінний автомобільний ліфт, на якому допускається переміщення автомобіля разом з людиною.

Сучасні автоматизовані паркувальні системи зустрічаються і під іншою назвою – автоматизовані паркінги. Весь процес переміщення автомобіля відбувається за допомогою механізмів і практично не вимагає присутності людини, тільки для ремонту в разі поломки.

Сам процес установки автомобіля на таку стоянку полягає в наступному – автомобіль заїжджає на платформу, водій включає гальмівну стояночну систему, або режим паркування, глушить двигун, потім залишає автомобіль і саму платформу, після чого натискає кнопку або використовує спеціальний пульт, сигнал від якого запускає рух платформи і транспортування автомобіля до



місця його стоянки. Процес повернення автомобіля з паркування відбувається в зворотному порядку.

Кожна паркувальна система складається з самого каркасу паркінгу, який є зовнішнім, тобто встановлений на вулиці, або ж внутрішнім – знаходиться в будівлі, і безпосередньо автоматичної паркувальної системи, що складається з сучасного, міцного, високотехнологічного паркувального обладнання, керованого комп'ютером.

Автоматизовані паркувальні системи бувають різних типів. Ті, що супроводжують автомобіль до місця паркування і ті, що самі є паркувальними платформами, на яких автомобіль знаходиться весь час стоянки.

Так само автоматизовані паркувальні системи поділяються на палетні і безпалетні [2].

Палета – платформа, на якій знаходиться автомобіль при його переміщенні. Раніше в паркінгах використовували тільки такий спосіб переміщення, але він виявив безліч мінусів. По-перше, після перенесення автомобіля на необхідне місце стоянки, наступний автомобіль можна завантажити тільки тоді, коли пуста палета повернеться на своє місце. Такий спосіб переміщення був не таким швидким, як хотілося б. Тому створили спеціальну «програму обміну палетами», щоб одна і та ж палета могла рухатися в різних напрямках і виконувати функції, як довозення автомобіля до місці парковки, так і його забору. Такий спосіб виявився зручнішим, але істотно підвищив вартість паркувальної системи за рахунок того, що істотна кількість енергії витрачається на те, щоб палети поверталися на місця не завантаженими. В результаті, відбувається безліч зайвих переміщень, при цьому не виконується жодна корисна робота. До того ж від різного розміру і ваги автомобілів, палети швидко зношуються.

В результаті чого були створені нові, безпалетні паркувальні системи, які представляють собою набір «пальцевих елементів», які переміщують автомобіль за допомогою гребінцевої технології обміну. Простіше кажучи, це спеціальні платформи – візки на колесах і рейках, які рухаються разом з

автомобілем, від місця до місця, залишаючись під ним на час паркування. Такі системи компактні, економічні, зручні, завжди готові перемістити автомобіль, не потрібно чекати, поки платформи приїдуть. У них немає «холостого» ходу без автомобіля, а значить і енергія не витрачається даремно.

Автоматизовані системи паркінгу (АСП) так само розрізняються за ступенем розміщення автомобілів залежно від конструкції приміщення. Розрізняють вертикальні, горизонтальні, поздовжні, поперечні і змішані АСП.

Автоматизовані паркувальні системи дозволяють істотно заощадити на місцях паркування автомобілів, а значить економічно і вигідно використовувати площу будь-якої будівлі чи приміщення. Паркування автомобіля відбувається без участі людини, значить він економить і свій час. Форма споруди для такого типу паркування може бути нестандартною, на відміну від класичних багаторівневих парковок, які диктують умови розташування, розміри і форму. Автоматизовані паркувальні системи можуть керуватися за допомогою спеціальних пультів, звичайних кнопок на панелях або навіть з допомогою магнітних карт чи смартфона [2,3].

Наявність таких способів управління дозволяє заблокувати доступ від парковок тим людям, для яких вони не призначені. Тобто в автоматичні паркувальні системи не можуть проникнути злодії, крадії автомобілів або вандали.

В будь-якому випадку такий паркінг дасть Вам можливість:

- Запроектувати гараж там, де для звичайного, рампового, немає місця.
- Знизити наведену на автомобіль площу з 15 метрів на машиномісце за класичного паркінгу в один поверх, до 1,63 метра квадратної площі землі на 1 машину при використанні автоматизованої вежі.
- Знизити висоту приміщення наведену на машиномісце.
- Знизити витрати на пожежогасіння і вентиляцію, піти від необхідності робити димовидалення, оскільки.
- З такого паркінгу немає необхідності робити евакуаційні виходи для водіїв.

## 1.1 Типові технічні рішення

Автоматизовані паркувальні системи - це індивідуальне, складне, інноваційне та сучасне обладнання, кожна така парковка розробляється і проектується на заводі під певне місце встановлення, тому систематизувати їх можна вельми умовно, існують рішення, які застосовують більш часто чи навпаки. Деякі типи паркувальних систем можуть бути схожі один на одного за схемою, але мати принципово різні способи переміщення автомобілів. Умовно можемо розділити їх на наступні великі групи - палетні і безпалетні, баштові і площинні, ті, що мають центральний проїзд для маніпулятора і ті, що займають усю площу поверху.

### 1.1.1 Механізовані парковки баштового типу

Баштовий паркінг, механізована парковка баштового типу (Tower Parking, паркінг «вежа») — це багаторівнева самонесуча конструкція, яка має центральний підйомник ліфтового типу з одно - або двухкоординатним маніпулятором [4].

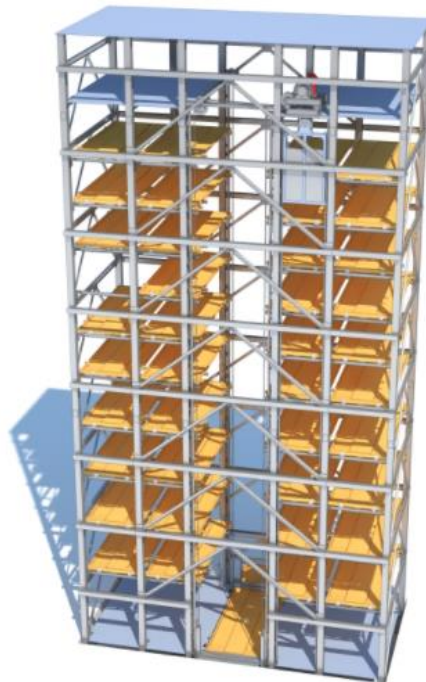


Рисунок 1.1 – Механізована парковка баштового типу. Проектний вигляд

Як правило, поздовжні або поперечні комірки для зберігання автомобілів у цьому багаторівневому паркінгу розташовуються з двох сторін від ліфтової шахти.

Можлива схема, в якій осередки знаходяться з чотирьох боків навколо шахти. Модульні паркінги баштового типу можуть мати зону прийому і видачі автомобілів, оснащену поворотним столом.

Паркувальні системи баштового типу застосовуються в тому випадку, коли мало місця для розміщення парковки іншого типу, але є місце для побудови її з вертикальною орієнтацією. Мінімальний габарит такої системи - 6,5-7,5 метрів, в даному випадку по центру розташований підйомник, який ставить автомобілі, як на полиці праворуч, так і ліворуч від себе, кількість машин обмежена нормами висотності будування парковок. Також існує можливість будувати вертикальну паркінгову систему, де автомобілі розташовуються по колу та з підйомником в центральній частині. Цей варіант більш місткий, але має більші габарити – його діаметр близько 20 метрів.

Але варіанти конфігурації форм парковки даного типу не обмежуються квадратом/прямокутником чи кругом, а розробляються індивідуально, залежно від можливості інсталяції та інших вимог [5].



Рисунок 1.2 – Механізована парковка баштового типу. Реальна забудова

Основна гідність баштового багаторівневого паркінгу — це мінімальна площа на один автомобіль. В середньому - 1 модуль займає площу 3-4 автомобілів. Тому сучасні паркінги баштового типу раціонально використовувати в місцях, де вкрай висока вартість землі. Тобто дані багаторівневі паркінги ефективно використовувати, в першу чергу, у великих мегаполісах типу Києва, Дніпра, Одеси та інших міст.

Принцип роботи. Для парковки автомобіля паркувальна система, як і всі автоматизовані паркінги, у роботі використовує принцип вертикального і горизонтального переміщення автомобілів, але із застосуванням технологічних процесів і механізмів, властивих тільки цьому типу паркувальних систем.

Паркування автомобіля до паркувальної системи проводиться наступним чином: водій підїжджає до воріт паркувального боксу, прикладає свою індивідуальну карту паркувального місця до зчитувача на панелі управління, або на складальному полі сенсорного екрану набирає комбінацію з цифр і букв, які ідентифікують його як власника, або особа, у якого є доступ до певного місця у гаражі [3, 4].

Ворота відкриваються, і водій вїжджає в бокс. Якщо паркувальна система палетного типу, водій вїжджає на паркувальну палету до тих пір, поки передні колеса не упруться в невеликий обмежувач. Якщо паркувальна система беспалетного типу, то водій вїжджає по напрямних так само, до упору коліс. Кожен паркувальний бокс оснащений комплектом певних датчиків і сенсорів, які при постановці автомобіля визначають його висоту, ширину, довжину, вагу.

Якщо по якихось параметрах автомобіль не відповідає допустимим габаритами, а так само, якщо водій встановив автомобіль не правильно, або яка частина автомобіля виходить за безпечний периметр, на світловому табло перед водієм висвітлиться покажчик, який вкаже, яка частина автомобіля випирає за безпечну зону. При цьому звукова сигналізація так само вкаже, що паркувати автомобіль при такій постановці заборонено. Якщо автомобіль встановлений неправильно, паркувальна система ніколи не почне процес паркування.

Водієві буде необхідно переставити автомобіль таким чином, щоб ні звукової, ні світловий сигнал не спрацьовували. Тільки після цього водій може покинути салон автомобіля і паркувальний бокс. Далі водієві необхідно знову з допомогою індивідуальної карти управління або за допомогою набірної поля на сенсорній панелі дати команду паркувальній системі виконати дію запаркувати автомобіль. Ворота закриваються, і автомобіль почне переміщення до свого паркувального місця.

Паркування автомобіля проводиться в кілька дій: перше – ліфт-маніпулятор починає переміщати автомобіль по вертикалі, друге – як тільки автомобіль досяг потрібного рівня, паркувальна система зміщує його в сторону, на паркувальне місце. Якщо застосовується автоматизована система паркування палетного типу, то шатл за допомогою спеціальних приводів перештовхне палету з автомобілем на паркувальне місце до спеціальних фіксаторів. Якщо ж застосовується паркувальна система беспалетного типу, то робот підніме автомобіль над ліфтом на 5 см, разом з автомобілем виїде на паркувальне місце і встановить автомобіль точно на передбачені для зберігання автомобіля підставки. Підставки мають фіксаційні виймки під колеса, тому автомобіль завжди надійно зафіксований на своєму паркувальному місці. Робот повертається на ліфт. Далі паркувальна система готова виконати наступний цикл.

### *1.1.2 Механізовані парковки, площинні, з проїздом по центру*

Це найпоширеніший тип механізованих паркувальних систем, він може бути багаторівневим і ідеально підходить для підземних і надземних гаражів, де необхідно збільшити кількість машиномісць у порівнянні зі звичайним паркінгом або не вистачає місця для організації проїзду для автомобілів з водієм. В даному випадку ширина проїзду обмежена розмірами автомобіля, так само машиномісця менші за габаритами і по висоті, можна поставити автомобілі в кілька рядів по боках від проїзду маніпулятора. Рівні, полиці на який ставляться машини, можуть бути виконані з бетону або металоконструкцій [1,6].

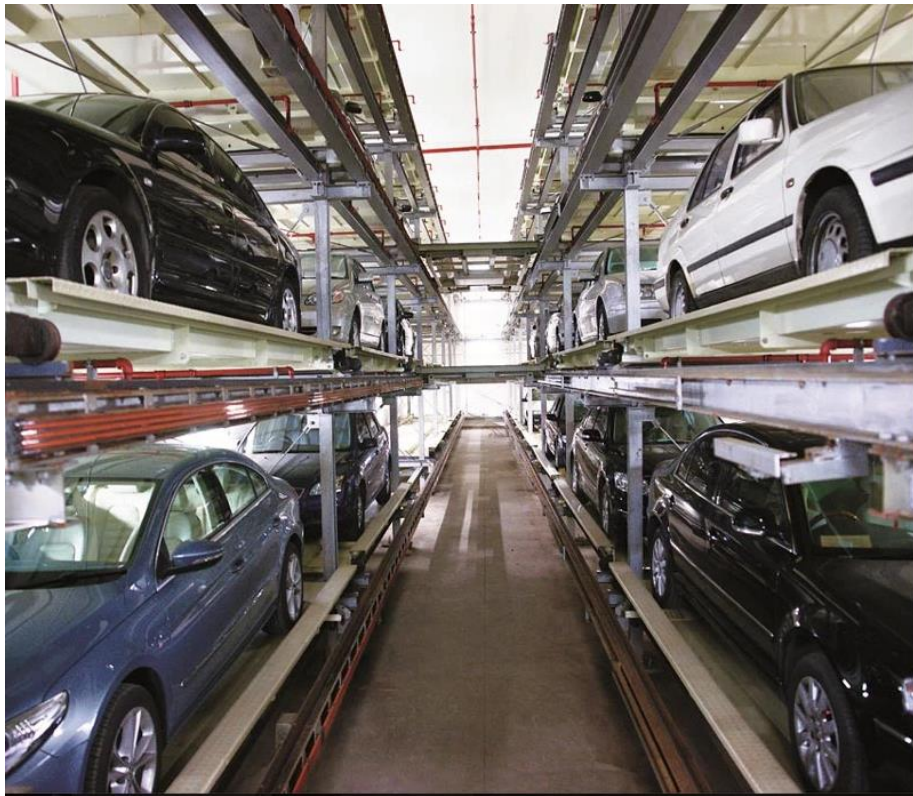


Рисунок 1.3 – Механізовані парковки, площинні, з проїздом по центру

#### *1.1.3 Механізовані парковки, площинні, без проїзду по центру*

Це самий економний по займаній площі варіант автоматизованого паркінгу, машиномісця займають всю площу паркування, залишивши одне місце для витягу і одне для перестановки машин, однак такий варіант не може бути використаний для великих або багаторівневих парковок, оскільки час подачі автомобіля при великій їх кількості буде занадто великим. В разі необхідності, можна зробити невеликий гараж в місцях, де обмежений доступний простір для будівництва. Це ідеальний варіант, наприклад, при постановці 20 автомобілів. Наведена площа може скласти всього 15 м. кв [1,2,6].





Рисунок 1.4 – Механізовані парковки, площинні, без проїзду по центру

#### *1.1.4 Підводні паркінги*

Йдеться про будівництво з використанням передових наукових досягнень і технічних можливостей повністю автоматизованих парковок і гаражів для автомашин під дном річок та каналів [7].

Ця пропозиція представляється цікавою, перш за все, з наступних причин:

- вирішення проблеми з виділенням майданчиків під будівництво парковок, особливо в центрі великих міст;
- очищення в прилеглий зоні дна річок при установці нижче рівня води повністю ізольованих резервуарів. Оздоровлення екології річок і каналів;
- скорочення термінів будівництва і експлуатаційних витрат на обслуговування парковок за рахунок відсутності витрат на опалення. Під дном річки температура постійно тримається на позначці 5-7 градусів тепла;
- ультрасучасні паркувальні ліфти, сполучні підземні парковки з набережними, які не вимагають багато місця і при грамотному рішенні можуть стати прикрасою архітектурного вигляду столиці чи інших міст;
- при діаметрі паркувального резервуара 20 м і при висоті піддонної конструкції 20 м можна розмістити 120 автомобілів. При збільшенні висоти - до



240 автомобілів. Швидкість обробки автомобіля від 60 до 90 сек. Таким чином, на одному кілометрі водної артерії можна розмістити до 12 000 одиниць легкового автотранспорту.

Такий пристрій повністю автоматизовано і управляється одним оператором. Від водія потрібно лише поставити машину в ліфт і вручити оператору магнітну картку [8].

Ліфт підніме автомобіль на потрібний ярус, а потім перемістить в осередок, відповідну коду картки. Восьми -, десятирівневі паркінги матимуть висоту до 40 м., у кожному з них можуть розміститися від 100 до 600 місць.

Поява підводних паркінгів додасть місту від 2,5 до 15 тис. машиномісць. Причому в місцях, де дефіцит автостоянок найгостріший.

#### *1.1.5 Паркувальна система BOX PARKING*

Паркувальна система «БОКС ПАРКНИГ» (BOX PARKING) – це інтелектуальна паркувальна система, призначена для багаторівневого постійного або тимчасового зберігання автомобілів.

Автоматизована паркувальна система «БОКС ПАРКНИГ» (BOX PARKING) може розміщуватися в підземних, наземних, або комбінованих гаражних комплексах і мати від 2-х до 8-и рівнів наземного розміщення і до 8-и рівнів підземного розміщення. За типом переміщення і в залежності від вимог або умов, паркувальна система може бути палетної (автомобілі переміщуються на індивідуальних паркувальних палетах) і беспалетной (автомобілі переміщуються на підставках під колеса). За типом зберігання та в залежності від вимог або умов, паркувальні рівні безпосереднього зберігання автомобілів паркувальної системи можуть бути виконані як з бетону, так і з металу. У варіанті з металевими рівнями для зберігання автомобілів беспалетный тип не застосовується в силу незахищеності нижніх автомобілів від потрапляння води і бруду з автомобілів, розташованих вище [9].

Відмінною особливістю автоматизованої даної паркувальної системи від інших, є її дуже компактне розміщення. Паркувальна система працює за

ротаційним принципом. У центральній частині розташовані місця для зберігання автомобілів, по краях розміщуються ліфтові пристрою. Паркувальні палети з автомобілями переміщуються по горизонталі, і коли палета з автомобілем виїжджає на ліфт, ліфт або піднімає на наступний рівень, або опускає на рівень нижче, і далі залишає автомобіль в загальному ряду. При цьому з іншого краю паркувальної системи відбувається те ж саме, але або палета з автомобілем опускається на рівень нижче, або піднімається. Відбувається ротація паркувальних місць до тих пір, поки палета з потрібним автомобілем не переміститься на ліфтове обладнання, яке потім піднімає її в паркувальний бокс.

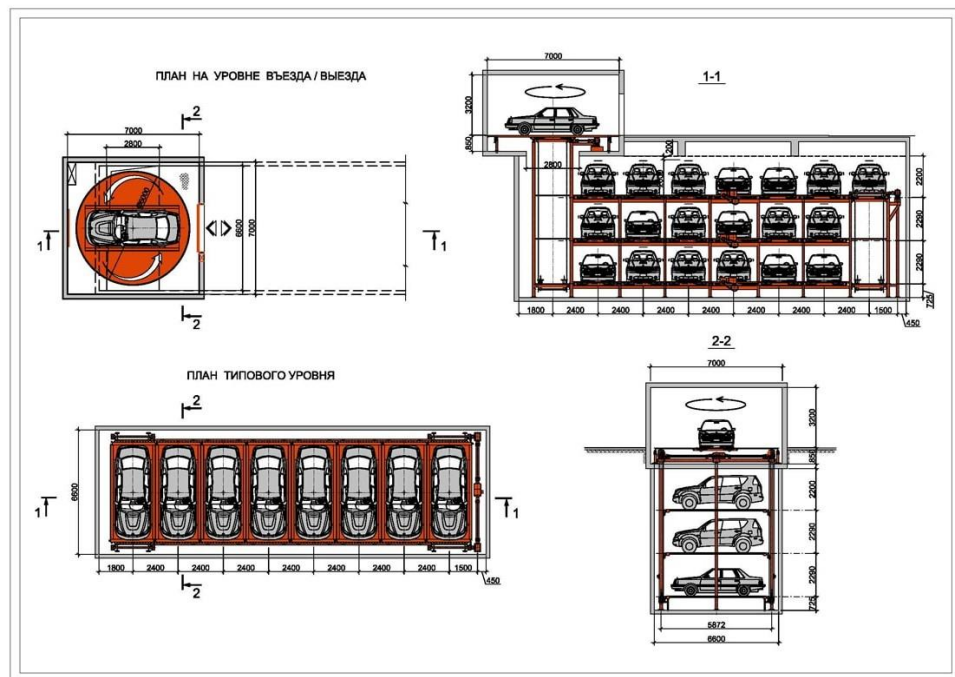


Рисунок 1.5 – Паркувальна система boxparking

#### 1.1.6 Паркувальний підйомник METRO 2P

Двостійковий паркувальний підйомник METRO 2P – призначений для паркування автомобілів у два рівня. Найкращий варіант для паркування автомобілів як в гаражних приміщеннях житлових будинків і адміністративних будівель, так і на відкритих ділянках. Простота установки і управління двостійковим паркувальним підйомником METRO 2P, а так само надійність, роблять його незамінним при бажанні без зайвих витрат ресурсів і часу отримати

додаткове паркувальне місце. Простий процес складання, компактне розміщення і просте управління за допомогою ключа/кнопок або брелок дистанційного управління роблять гідравлічний паркувальний підйомник METRO 2P доступним для всіх груп користувачів [10].

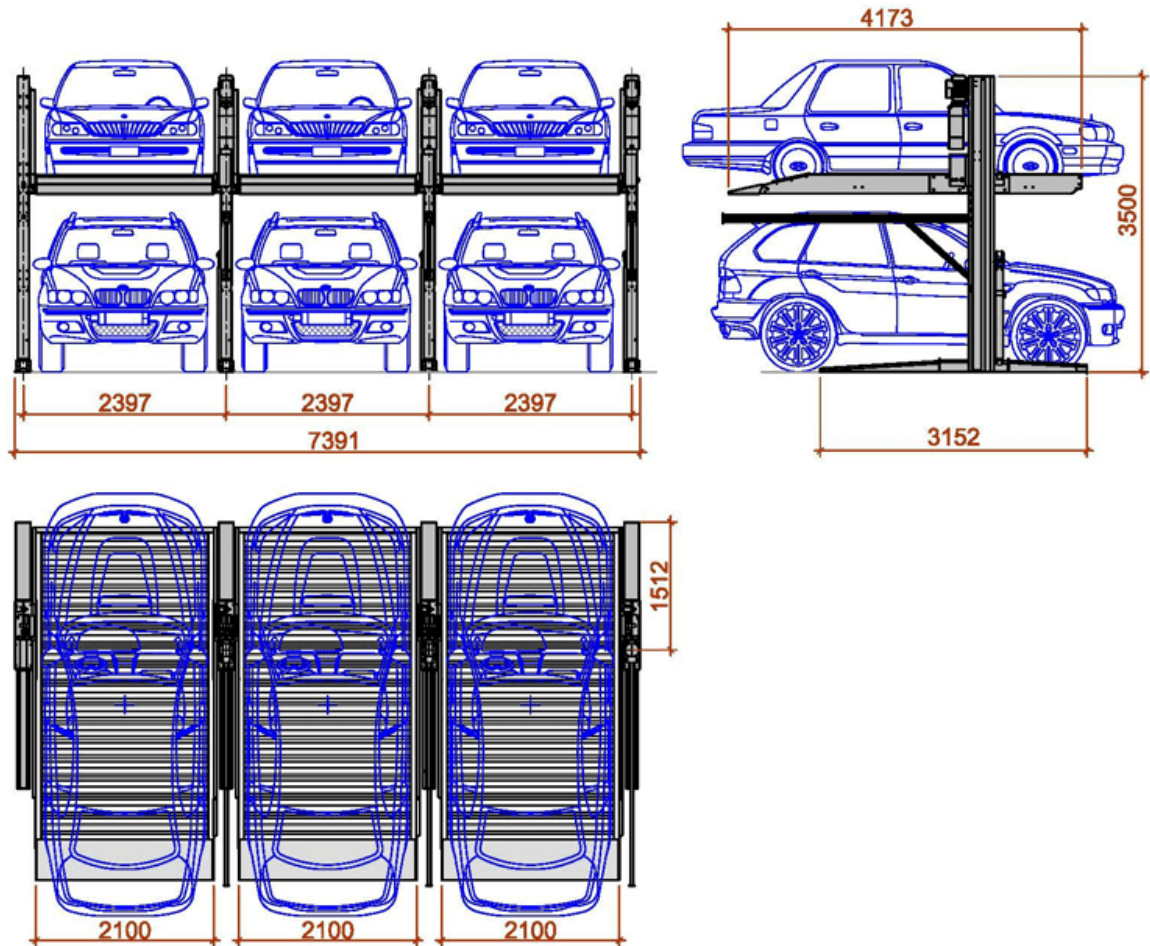


Рисунок 1.6 – Паркувальний підйомник METRO 2P

#### 1.1.7 Паркувальна система ТРАНСЛО ПАРКІНГ

TRANSLO PARKING – це інтелектуальна паркувальна система, призначена для багаторівневого постійного або тимчасового зберігання автомобілів. Автоматизована система паркування може розміщуватися в підземних, наземних, або комбінованих гаражних комплексах і мати від 1-го до 9-ти рівнів розміщення [11].

Відмінною особливістю даної автоматизованої паркувальної системи від інших систем, є ліфт-маніпулятор, який поєднує в собі і ліфтовий пристрій і маніпулятор горизонтального переміщення. Ліфтовий пристрій переміщається по

горизонталі в центральній частині гаражного приміщення по рейках з одночасним переміщенням маніпулятора по вертикалі. При роботі автоматизованої паркувальної системи відбувається дві дії: перше – ліфт-маніпулятор переміщається по рейках, друге – переміщення маніпулятора на ліфті вгору/вниз. При комбінації цих двох дій, автомобіль з паркувального боксу переміщується до свого паркувального місця по діагоналі, тобто самому короткому шляху з точки А в точку Б.

За рахунок того, що автомобіль в даній автоматизованій паркувальній системі переміщається по діагоналі до свого паркувального місця, автоматизована система є однією з найбільш швидкісних паркувальних систем в лінійці автоматизованих паркінгів. Додатково – це значно скорочує час на виїзд автомобіля з паркувального боксу і істотно спрощує процедуру. Ліфтові пристрої оснащуються розворотними механізмами, які повертають автомобіль на  $180^\circ$  і дозволяють отримувати автомобіль капотом на виїзд. Це істотно скорочує час виїзду автомобіля з паркувального боксу і знижує вплив людського фактора на продуктивність паркувальної системи з точки зору кількості операцій на годину, а так само робить процес виїзду з паркувального боксу значно простішим і комфортнішим.

У випадку, якщо передбачається гаражний комплекс з великою кількістю паркувальних місць і рівнів, рекомендується установка декількох ліфтових пристроїв з декількома паркувальними боксами. Таке рішення дозволяє одночасно проводити дії з декількома автомобілями, що забезпечить високу продуктивність паркувальної системи, дозволить уникнути виникнення черг та заторів, дозволить більш ефективно використовувати паркувальну систему в години високих навантажень.

Паркування автомобіля паркувальною системою проводиться в кілька дій: перше – ліфтовий пристрій везе маніпулятор з автомобілем рейкою по горизонтальній площині, друге – одночасно з цим маніпулятор з автомобілем переміщається на рухомому ліфті у вертикальній площині. Автомобіль переміщається в просторі паркінгу по діагоналі від паркувального боксу до свого

паркувального місця. Напроти необхідного паркувального місця ліфт-маніпулятор зупиняється і переміщує автомобіль на паркувальне місце.

Паркування автомобіля до паркувальної системи займає в середньому 1,5-2 хвилини.

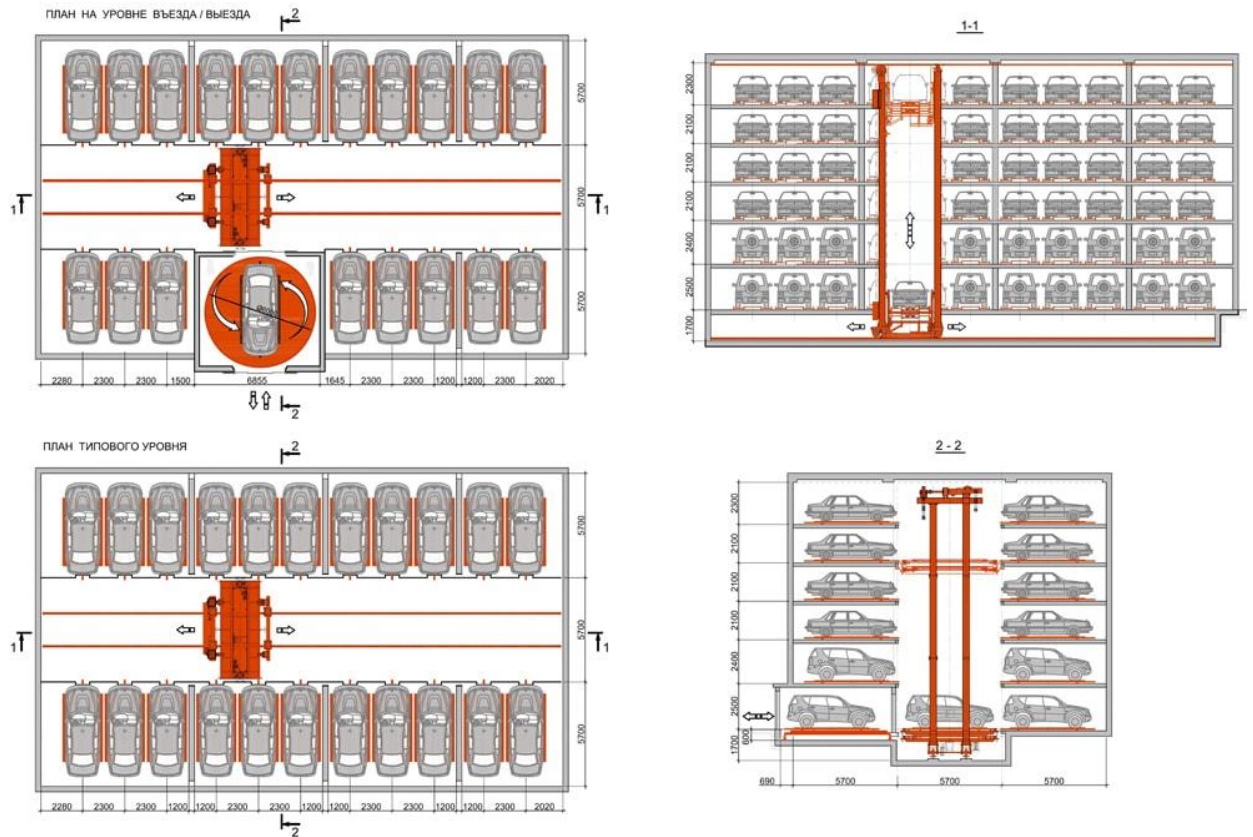
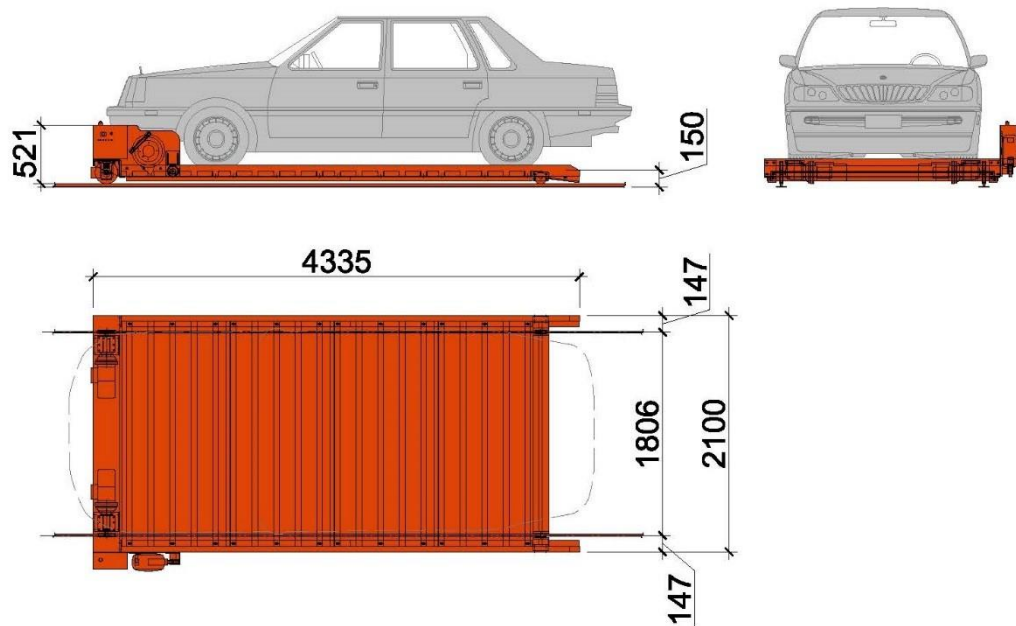


Рисунок 1.7 – Паркувальна система ТРАНСЛО ПАРКІНГ

#### 1.1.8 Паркувальна платформа ТТ/ТЛ

Паркувальна платформа ТТ/ТЛ – призначена для паркування автомобілів у гаражах з низькою висотою стелі. Особливістю паркувальної платформи ТТ/ТЛ є її здатність переміщатися по рейках, встановлених на підлозі гаража. За рахунок переміщення паркувальної платформи з автомобілем вперед-назад або вправо-вліво, з'являється можливість переміщати припаркований автомобіль, відкриваючи виїзди для автомобілів, перед якими платформа встановлена. Єдино можливий варіант для паркування автомобілів в гаражних приміщеннях житлових будинків і адміністративних будівель з низькою висотою стель. Простота монтажу та управління паркувальної платформою ТТ/ТЛ, а так само її надійність, роблять її незамінною при бажанні без зайвих витрат

збільшити кількість паркувальних місць. Простий процес монтажу, компактне розміщення і просте управління за допомогою ключа/кнопок роблять паркува-



льну платформу TT/TL доступним для всіх груп користувачів [12].

Рисунок 1.8 – Паркувальна платформа TT/TL

## 1.2 Опис електромеханічної системи багатоярусної паркінгової системи автомобілів

### 1.2.1 Опорні стійки

Застосовувана при виробництві опорних стійок сталь, забезпечує подвійний запас міцності конструкції, здатна витримати так само і подвійні динамічні навантаження. Кріплення опорних стійок до поверхні підлоги здійснюється 16 анкерними болтами, що повністю виключає зміщення або розгойдування стійок або підйомника. Суцільна проварка швів конструкції так само забезпечує необхідну ступінь жорсткості і надійність кріплення деталей підйомника. Забарвлення опорних стійок проводиться по технології порошкового фарбування і полягає в електростатичному розпилення фарбувальної суспензії на метал, яка рівномірно сідаючи на металеву поверхню крім безпосередньо самої забарвлення робить захист металу стійкою до різних агресивних середовищ (бензин, масла, реагенти) і механічних впливів.



### *1.2.2 Паркувальна палета*

Паркувальна палета складається з чотирьох основних балок, які формують основний каркас палети, і фіксуючих між собою замковим способом металевих гофрованих листів, становлять робочу поверхню паркувальної палети. Основні балки палети кріпляться між собою болтами, що забезпечує надійне кріплення загального контуру палети, надає необхідну жорсткість кріплення і запобігає перекосам палети при експлуатації. Робочу поверхню паркувальної палети виконано у виді гофрованого листа, що так само забезпечує жорсткість поверхні і достатній опір навантаженням, запобігаючи від прогинання і деформацій загального полотна палети. Для захисту від агресивних середовищ при експлуатації, палета оброблена оцинкуванням виконаної гарячим способом, що значно збільшує ступінь захисту і термін служби робочої поверхні палети.

### *1.2.3 Система синхронізації*

Система синхронізації механізму підйому паркувальної платформи покликана забезпечувати плавний, строго горизонтальний підйом паркувальної платформи паркувального підйомника. Даний механізм складається з набору шестерень і ланцюгів, які з'єднані з гідравлічними циліндрами і при висуванні штока гідроциліндра забезпечують рівномірність висування штоків, так і вирівнювання паркувальної платформи в разі нерівномірності навантаження від неправильно встановленого автомобіля на паркувальну палету.

### *1.2.4 Електрогідравлічний блок*

Електрогідравлічний блок складається з:

- 1.2.1. Електродвигуна;
- 1.2.2. гідравлічної помпи з регулювальними клапанами (клапан перевищення максимального тиску, направляючий клапан електромагнітний клапан, видатковий клапан);
- 1.2.3. бака для гідравлічної рідини;
- 1.2.4. гідравлічних циліндрів;
- 1.2.5. гідравлічних шлангів.

Електрогідравлічний блок покликаний забезпечити паркувальний підйомник необхідної підйомної енергією, необхідної для процесу підйому і опускання паркувальної палети з автомобілем. Даний процес досягається за рахунок перетворення електричної енергії в крутний момент ротора, далі ротор двигуна передає крутний момент на гідравлічну помпу, яка в свою чергу створюючи тиск в гідравлічному контурі приводить в рух шток гідравлічного циліндра, який, виштовхуючись з корпусу, піднімає паркувальну платформу з автомобілем.

#### *1.2.5 Шафа електрична*

Електрична шафа підйомника складається з набору автоматичних вимикачів (трьох - і однофазні) , 6 проміжних реле, реле часу, трансформаторів 220В і 24В, запобіжників, контактора та діодного моста, що забезпечують процес розподілу електричної енергії від вхідної напруги кабелю між виконуючими пристроями підйомника.

На центральній панелі шафи розташовані елементи управління (поворотний важіль включення/вимикання живлення, світловий індикатор наявності живлення в системі і звуковий сигналізатор наявності сторонніх предметів або автомобіля в процесі опускання паркувальної палети).

#### *1.2.6 Панель управління*

Панель управління підйомником виконана в виносному варіанті і встановлена на штанзі. Даний варіант дозволяє користувачеві перебувати на безпечній відстані від паркувальної палети з автомобілем у момент підйому/спуску. З точки зору безпеки – це найкращий варіант, при якому користувач стежить за процесом, при цьому перебуваючи на безпечній відстані від рухомих елементів.

Панель управління завжди забезпечена 3-ма елементами управління:

— Поворотний ключ / поворотна кнопка-важіль/кнопки підйому-опускання;



- кнопка аварійної зупинки;
- світловий індикатор.

Так-само можливі варіанти розміщення пультів управління паркувальним підйомником на колонах і стінах.

Як опція можливе виготовлення пультів управління паркувальним підйомником дублюванням управління через брелок дистанційного керування. Наявність дистанційного управління дозволяє опустити паркувальну платформу для паркування не виходячи з автомобіля, що суттєво скорочує загальний час паркування і робить його більш комфортним. Паркувальну палету так само значно комфортніше і безпечніше піднімати натискаючи на кнопку на брелоку дистанційного керування.

### **1.3 Формулювання вимог до систем автоматичного керування багаторушної паркінгової системи автомобілів**

Для повноцінного функціонування системи автоматизації багаторушної паркінгової системи для автомобілів необхідно виконати ряд наступних вимог, що представлено нижче [13].

- Підвищення якості функціонування об'єкта управління;
- Підвищення якості управління та інше;
- Паркування автомобіля без водія;
- Компактне розміщення автомобілів;
- Відсутність пандусів для в'їзду у гаражне приміщення;
- Найвищий рівень безпеки при парковці і зберіганні автомобілів;
- Мінімальний час паркування або очікування автомобіля;
- Високий рівень комфорту при користуванні паркінгом;
- Пульт управління платформою в безпечній зоні;
- Управління підйомником будь-яким користувачем;

- Надійність АСУ в цілому і кожної її автоматизованої функції повинна бути достатня для досягнення встановлених цілей функціонування системи при заданих умовах застосування;
- Адаптивність АСУ повинна бути достатньою для досягнення встановлених цілей її функціонування в заданому діапазоні змін умов застосування;
- В АСУ повинні бути передбачено контроль правильності виконання автоматизованих функцій і діагностування, з зазначенням місця, виду і причини виникнення порушень правильності функціонування АСУ;
- В АСУ повинні бути передбачені заходи захисту від неправильних дій персоналу, що призводять до аварійного стану об'єкта або системи управління, від випадкових змін і руйнування інформації і програм, а також від несанкціонованого втручання;
- АСУ в необхідних обсягах повинна автоматизовано виконувати: збір, обробку і аналіз інформації (сигналів, повідомлень тощо) про стан об'єкта управління; вироблення керуючих впливів (програм, планів тощо); передачу керуючих впливів (сигналів, відомостей тощо) на виконання та її контроль; реалізацію та контроль виконання керуючих впливів; обмін інформацією (документами, повідомленнями тощо) з взаємопов'язаними автоматизованими системами;
- Технічні засоби АСУ, що використовуються при взаємодії АСУ з іншими системами, повинні бути сумісні по інтерфейсах з відповідними технічними засобами цих систем і використовуваних систем зв'язку.

### **Висновки до розділу**

1. Виконано огляд паркінгових систем, в яких автовласник не заходить в паркінг, а просто ставить автомобіль в бокс і йде, система паркування шукає місце, пересуває, розгортає і потім сама видає назад автомобіль.
2. Водій може поставити і викликати машину з системи не тільки за картою або номером на дисплеї, але і за допомогою спеціальної програми на смартфоні або телефонним дзвінком і коли він підходить до боксу його машина вже на місці.
3. На підставі виконаного огляду обрано багатоярусну паркінгову систему баштового типу для проєктування і дослідження в роботі.

## РОЗДІЛ 2. ВИЗНАЧЕННЯ ПАРАМЕТРІВ ЕЛЕКТРОМЕХАНІЧНОЇ СИСТЕМИ БАГАТОЯРУСНОЇ ПАРКІНГОВОЇ СИСТЕМИ АВТОМОБІЛІВ

### 2.1 Прототип конструкції проектного стенду

Далі представлено опис вузлів паркувальної системи згідно рисунків-візуалізації одного з реально діючих варіантів побудови даного типу паркінгу. Візуалізація створена для ознайомчих цілей, та може не враховувати певні деталі, що не впливають на загальний функціонал системи [14].

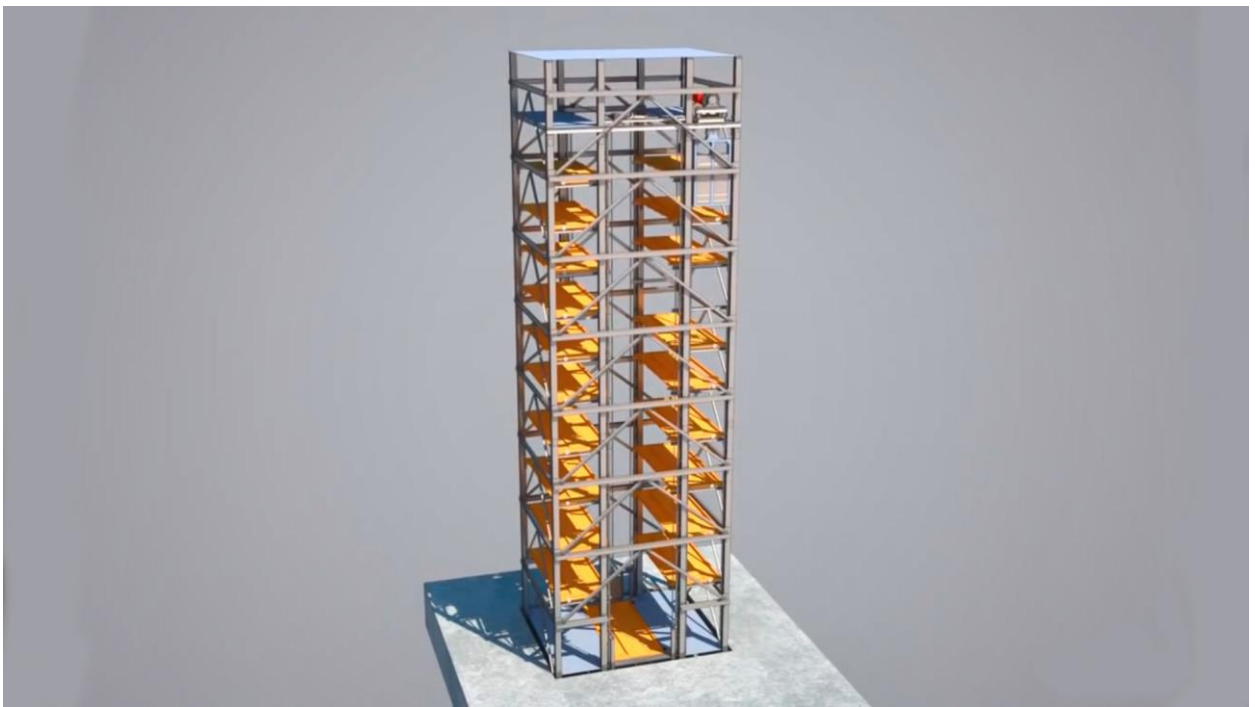


Рисунок 2.1 – Зовнішній вигляд багатоярусної паркінгової системи баштового типу

Багатоярусна паркінгова система баштового типу, взята за основу дипломного проекту, має класичний вигляд стандартної прямокутної форми і буде вміщувати від 10 до 20 автомобілів (від 5 до 10 ярусів). Конструкція складається із металевих балок, які виконують несучу функцію, перегородок для покращення жорсткості, та, власне, ліфтового механізму, який, в свою чергу, складається із підйомного механізму, протизаги, поворотного механізму, штохачі (для горизонтального переміщення палети).

Окремо передбачується планшет для керування системою через Android-додаток. На візуалізації його не зображено, так як попередньо під планшетом мається на увазі встановлений додаток на смарт-пристрої користувача.

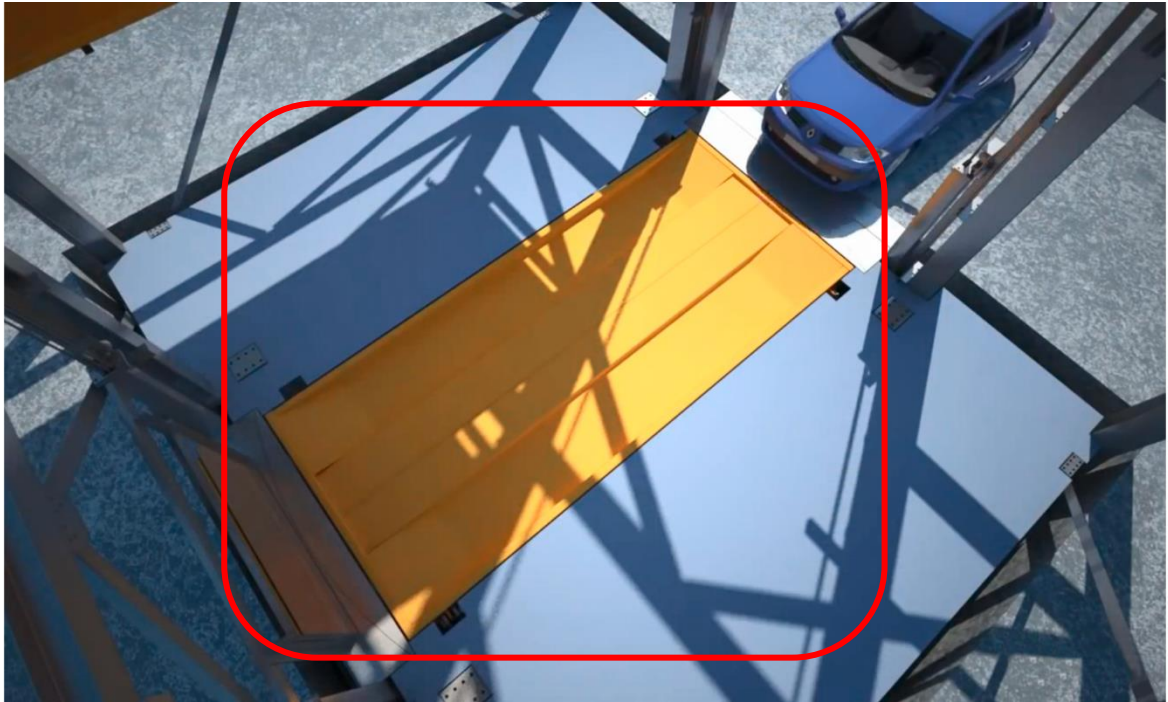


Рисунок 2.2 – Зовнішній вигляд палети. Початок процесу завантаження авто

На рисунку вище зображено палету, яка використовується для утримання автомобіля на рухомій платформі (ліфті). Палета обладнана спеціальними впадинами для перешкодження самовільного відкочування автомобіля із коректної, безпечної позиції під час переміщення.



Рисунок 2.3 – Механізм підйому, ролики розподілу моменту

На рисунку вище зображено ліфтовий механізм з боку розміщення роликів, мета яких рівномірно розподіляти момент від двигуна та противаги за рахунок вільного ходу.



Рисунок 2.4 – Противага

Процес вирівнювання моменту необхідний для забезпечення стабільної роботи ліфтового механізму, зокрема перешкоджанню виникненню перекосів



на напрямних. Противага виконує роль демпфера, що забезпечує плавність роботи системи переміщення, та полегшує роботу двигуна зокрема при пуску і під час робочого циклу, забираючи частину навантаження на себе.

На рис. 2.5 зображено розміщення поворотного механізму палети, котрий призначений для розвертання автомобіля під час видачі користувачу на 180 градусів, для комфортного виїзду із приймального боксу. Таким чином, користувач під час паркування заїжджає в приймальний бокс обличчям вперед, а при отриманні автомобіля система розвертає його, для такого ж зручного та швидкого виїждження із боксу.



Рисунок 2.5 – Поворотний механізм. Вид знизу

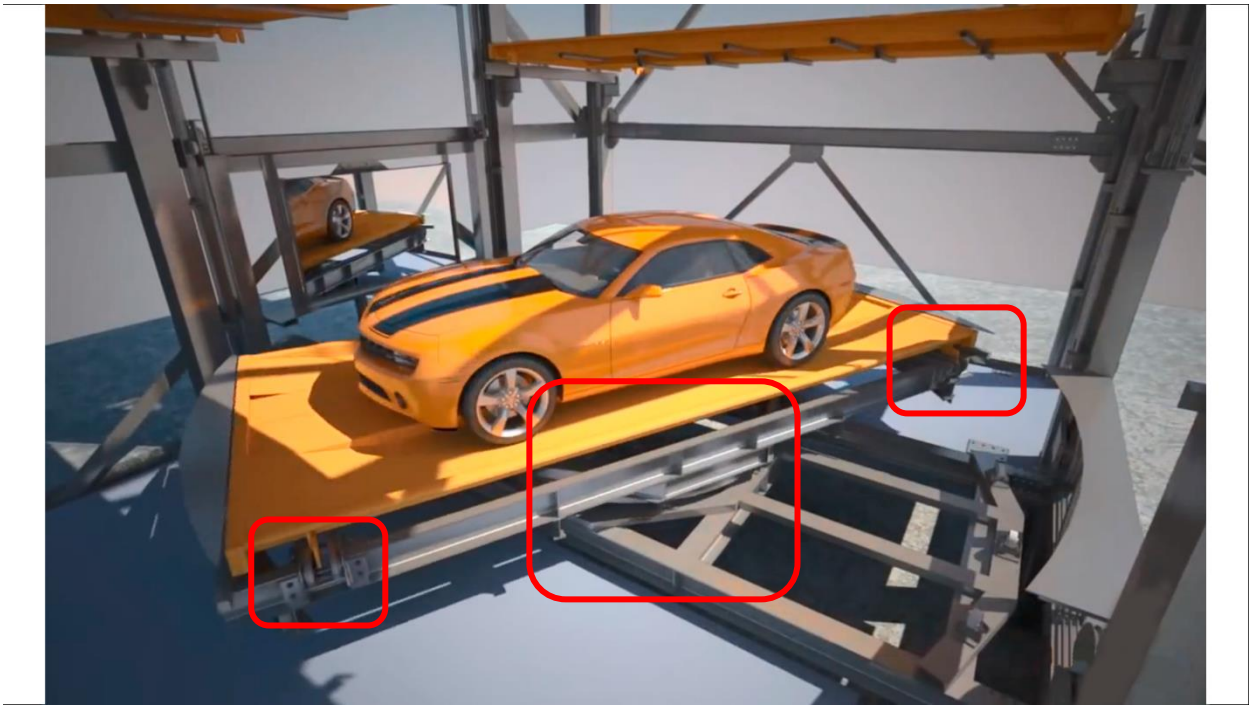


Рисунок 2.6 – Поворотний механізм, в процесі розвороту. Штовхальний механізм

Як видно з рис. 2.6, палета із автомобілем повністю розміщена на поворотній осі із поворотним механізмом. Це висуває особливі вимоги до надійності та збалансованості конструкції, окрім того ряд запобіжних засобів, котрі не дозволяють випадково запустити процес повертання авто ні в жодному випадку, окрім як передбаченого сценарно.

Також на рис. 2.6 можна розгледіти штовхальний механізм, основною задачею якого є завантаження палети на головну стійку для зберігання авто та подальше розвантаження стійки при запиті від користувача.



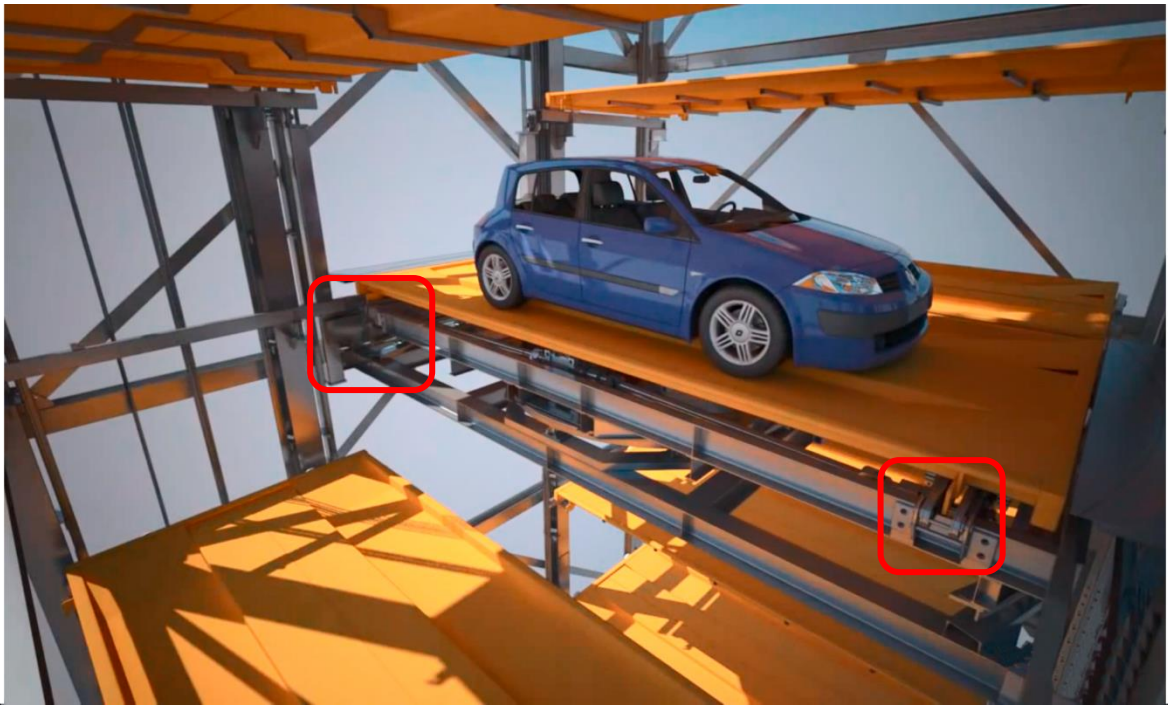


Рисунок 2.7 – Початок процесу завантаження авто на головну стійку

На рис. 2.7 зображено момент закінчення позиціонування системи на заданій комірці парковки, та початок процесу завантаження автомобіля клієнта на головну стійку. Описаний вище штовхальний механізм починає розкривання шасі для переміщення палети на спеціальні зачепи. На палеті передбачено набір роликів для зручного переміщенню її по напрямних, їх зовнішній вигляд очевидніший після повного відкриття шасі палети (див. рис. 2.8).

На рис. 2.8 детально видно ролики палети, більш розгорнуто зображено процес відкриття шасі палети. Окрім того, можна спостерігати за конструкцією головної стійки, на яку виконується завантаження палети з автомобілем. В цілому, вона складається із міцного профілю з ребрами жорсткості, та спеціальними демпферними накладками для м'якого ковзання роликів палети, що позитивно впливає на термін експлуатації останніх.



Рисунок 2.8 – Вид збоку. Процес забору авто з паркомісця

## 2.2 Огляд електротехнічної складової проектного стенду

### 2.2.1. Електропривідна частина

На зараз розглядається декілька типів двигунів, котрі можливо використати в стенді. Кожен з них має свої особливості керування та специфіку використання в даному проекті. Один з цих варіантів – кроковий двигун NEMA 17, тип 17HS4401. Кроковий електродвигун - це синхронний бесщітковий електродвигун з декількома обмотками, в якому струм, що подається в одну з обмоток статора, викликає фіксацію ротора. Послідовна активація обмоток двигуна викликає дискретні кутові переміщення (кроки) ротора.

Застосовуються крокові електродвигуни Nema 17 в приводах машин і механізмів, що працюють в старт-стопному режимі, або в приводах безперервного руху, де керуючий вплив задається послідовністю електричних імпульсів, наприклад, в верстатах з ЧПУ, 3D принтерах, лазерних граверів або верстатах плазмового різання. Для керування двигуном необхідно використовувати спеціальний драйвер, наприклад TB67S109AFTG. Цей драйвер позиціонується виробником як аналог модулів на популярному чіпі TB6600.

Для управління драйвером використовується три лінії - STEP (крок), DIR (напрямку обертання) і ENABLE (включення силової частини драйвера S109AFTG). При необхідності драйвер може працювати в режимі мікрокроку. Режим мікрокроку ( $1/2$ ,  $1/4$ ,  $1/8$ ,  $1/16$ ,  $1/32$  кроку) і струм через обмотки (від 0.5 до 3.5 А) встановлюється за допомогою групи перемикачів SW1 ... SW6.

Використання такого типу двигуна є найбільш простим, не враховуючи складнощів керування та необхідності додаткового обладнання, але не є автентичним. Тому перевага надається двигуну постійного струму. Зазвичай двигуни постійного струму не великих габаритів мають дуже велику швидкість обертання, що створює необхідність або в роботі двигуна на мінімальній швидкості за рахунок ШІМ, але такий спосіб значно зменшить енергоефективність системи, або встановленні редуктора, котрий знизить оберти двигуна до прийнятних. Також двигун має мертву зону керування, при якій ще не здатний виконувати корисну роботу, але вже споживає енергію.

При навантаженні ця зона розширюється. Експериментальним способом буде встановлено градацію ефективності роботи двигуна та внесено програмні обмеження на величину сигналів керування.

ДПС також потребує драйвера для підсилення слабострумівих сигналів з контролера. Модуль на мікросхемі драйвера L293D призначений для управління кроковими і колекторними двигунами невеликої потужності. Модуль виконаний в компактному виконанні і містить всі необхідні компоненти і сигнали управління для повноцінного управління індуктивним навантаженням.

Найбільш популярні застосування модуля - самохідні та радіокеровані іграшки, системи поливу, управління соленоїдами і електромагнітними двигунами і т.д. Живлення логічної частини можна здійснювати як від вхідної напруги силової частини (VIN), так і від окремого джерела живлення 5В (VCC). При живленні модуля від зовнішнього джерела (7-20В), можна використовувати внутрішній стабілізатор модуля 5В для живлення керуючого мікроконтролера Arduino. Також на модулі є роз'єм 5-пін для підключення крокового двигуна 28BYJ-48.

Переваги та недоліки ДПС [15]..

Переваги:

- простота будови та управління;
- практично лінійні механічна і регульовальна характеристики двигуна;
- легко регулювати частоту обертання;
- хороші пускові властивості (великий пусковий момент), (найбільший пусковий момент у ДПС з послідовним збудженням);
- компактніше інших двигунів (якщо використовувати сильні постійні магніти в статорі);
- оскільки ДПС є оборотними машинами, з'являється можливість використання їх як в руховому, так і в генераторному режимах.

Недоліки:

- дорожнеча виготовлення;
- для живлення електродвигуна від мережі змінного струму необхідно використовувати випрямні пристрої;
- необхідність профілактичного обслуговування колекторно-щіткових вузлів;
- обмежений термін служби через зношення колектора.

### 2.2.2 Контролер

Вибір контролера не однозначний, оскільки є можливість використання більш дешевого обладнання Arduino Mega 2560 R3, яке повноцінно справиться із поставленою задачею, натомість одноплатний комп'ютер Raspberry Pi 3 B+ має значно вищу вартість, але і не зрівнянно вищі можливості до обчислень. Якщо оцінити об'єктивно – потенціал більш потужного пристрою не буде реалізований в даному проекті, тому його використання не є обґрунтованим ні економічно ні технічно [16, 17]..

Raspberry Pi побудований на системі-на-чипі (SoC) Broadcom BCM2835, яка включає в себе процесор ARM із тактовою частотою 700 МГц, графічний

процесор VideoCore IV, і 512 чи 256 мегабайтів оперативної пам'яті. Твердий диск відсутній, натомість використовується SD карта.

Молодша (A) і старша (B) моделі Raspberry Pi відрізняються об'ємом оперативної пам'яті (256 і 512 мегабайтів відповідно) і кількістю USB-портів (один проти двох). Крім цього, у старшої моделі є роз'єм Ethernet 10/100, а молодша споживає на третину менше енергії [18]..

Для Raspberry Pi випущено спеціалізований дистрибутив Linux, Raspbian OS (заснований на дистрибутиві Debian) і магазин застосунків Pi Store, де є як платні, так і безплатні програми. Серед іншого Raspberry Pi використовує веб-браузер Iceweasel та KOffice, які постачаються в комплекті. Офіційна мова програмування для навчання — Python [19]..

Arduino Mega 2560 R3 є просунутою версією плати Arduino. Вона побудована на базі мікроконтролера ATmega2560. Як USB-UART перетворювач використовується мікросхема ATmega16U2. На борту Mega 2560 R3 є 54 цифрових входи/виходи (15 з яких можуть використовуватися як виходи ШІМ), 16 аналогових входів, 4 апаратних порти UART, тактуючий кварц на 16 МГц, USB коннектор, роз'єм живлення, кнопка RESET і роз'єм для внутрішньосхемного програмування ICSP. Arduino Mega 2560 R3 сумісна з більшістю плат розширення, розроблених для платформ Uno або Duemilanove. Дана плата - оптимальний вибір для реалізації проектів, які потребують великого обсягу пам'яті, а також в разі необхідності підключення великої кількості периферійних пристроїв.

### *2.2.3 Датчики*

В системі планується використовувати декілька індуктивних датчиків приближення для визначення крайніх положень рухомої частини парковки (підйомного механізму - ліфта) та визначення необхідного рівня (поверху) для проведення подальших маніпуляцій. Первинно вибір зупинився на індуктивному датчику приближення TL-W5MC1. Індуктивний NPN датчик TL-W5MC1 призначений для визначення наближення до датчика металевих поверхонь.

Часто такі датчики застосовують в системах автоматизації як кінцеві датчики положення. Живиться датчик від напруги постійного струму від 6 В до 36 В. Вихід - нормально розімкнутий [20].

### **Висновки до розділу**

1. В цьому розділі описано складові проєкту електромеханічної системи багатоярусної паркінгової системи автомобілів з умовно загальним поясненням деталей, оскільки існує декілька можливих варіантів реалізації, кожен з яких має свої переваги та недоліки.
2. Передбачено ряд засобів для більш плавної, тихої та надійної роботи системи, в тому числі наявність противаги та демпферів, плавного регулювання силовим агрегатом (двигуном), можливе встановлення редуктора.
3. Можливе встановлення двох типів двигунів, варіанти не є взаємозамінними.
4. Можливе використання двох типів контролерів, але більш доцільний - Arduino Mega 2560 R3 через свою вартість, наявність широкого клубу прихильників, відомих методів та продуктивності, яка абсолютно задовольняє потреби проєкту.

## **РОЗДІЛ 3. ВИБІР ЕЛЕКТРОДВИГУНІВ ТА ЕЛЕКТРОННИХ КОМПОНЕНТІВ. РОЗРОБКА (ОПИС) СХЕМИ**

### **3.1 Вибір та опис двигуна**

Фінальний вибір залишається на стороні мотор-редуктора постійного струму типу 12GA-12-400. Мініатюрний мотор-редуктор постійного струму з міцними і надійними сталевими шестернями. Маленькі габарити, легкий вага і високий крутний момент на низьких обертах. На вал двигуна легко можна встановити зубчасте колесо [21,22].

Редуктор - це пристрій, призначений для перетворення високої швидкості обертання вхідного вала в більш низьку швидкість вихідного валу. Мотор-редуктор - це конструкція, яка утворюється, якщо з'єднати редуктор і електро-двигун. Застосовується там, де потрібно витримування високих навантажень у поєднанні з чималим ККД. Такі мотори мають широке застосування:

- в робототехніці;
- в дитячих іграшках;
- у побутовій техніці;
- в автомобілебудуванні;
- в периферійних пристроях;
- у медичному обладнанні.

Переваги:

- компактний дизайн;
- низький момент інерції;
- експлуатація в будь-яких монтажних положеннях;
- великий крутний момент при мінімальних габаритних розмірах;
- застосовуються для тривалих, повторно-короткочасних і реверсивних режимів роботи [23].

Технічні характеристики обраного мотор-редуктора вказані в табл. 3.1.

Таблиця 3.1 – Технічні дані двигуна типу 12GA-12-400

Назва параметру	Значення параметру
Номінальна напруга	DC 12 В
Швидкість обертання без навантаження	400 об/хв
Швидкість обертання під навантаженням	304 об/хв
Номінальний струм навантаження	200 мА
Крутний момент (ном.)	0,3 кг/см
Діаметр вала	3 мм
Довжина валу	10 мм
Форма вала	D-подібний
Розміри мотора	28x12x10 мм.
Вага	10 грам
Матеріал шестерні	метал

### 3.2 Вибір драйверу для керування мотор-редуктором

Драйвер L298N використовується радіоаматорами для багатофункціонального управління двигунами постійного струму. Схема модуля, що складається з двох H-мостів, дозволяє підключати до нього один біполярний кроковий двигун або одночасно два щіткових двигуна постійного струму. При цьому є можливість змінювати швидкість і напрямок обертання двигунів. Управління здійснюється шляхом подачі відповідних сигналів на командні входи, виконані у вигляді штирьових контактів. На рис. 3.1 показаний зовнішній вигляд модуля з коротким описом всіх його складових [24,25].



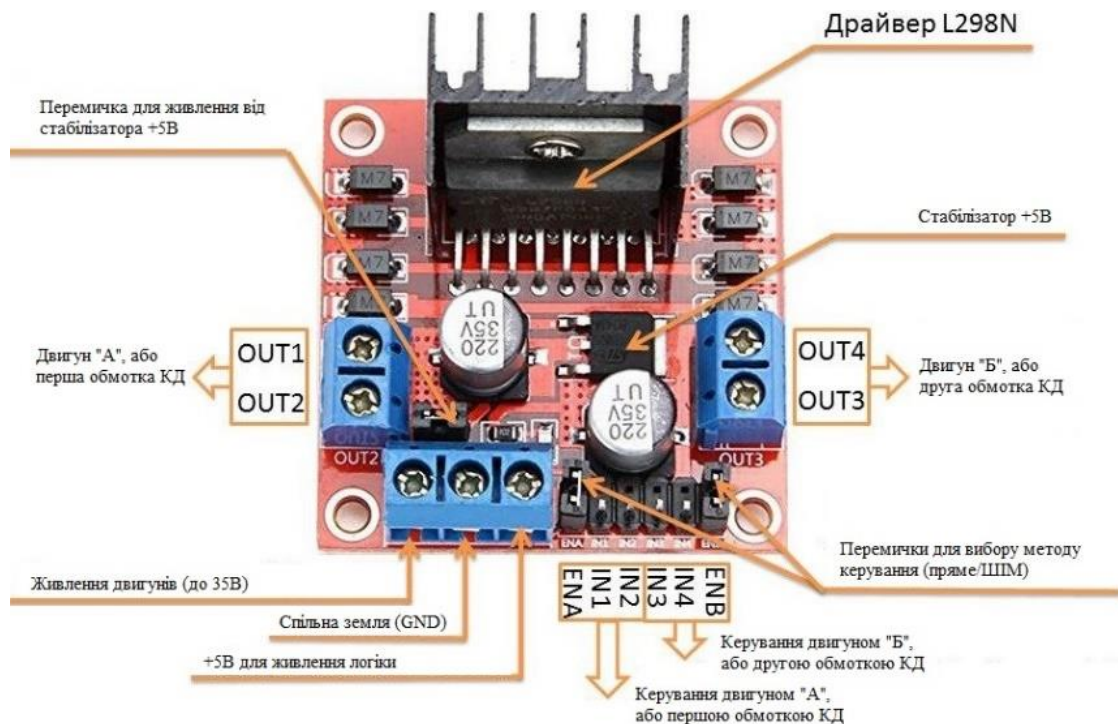


Рисунок 3.1 – Зовнішній вигляд драйвера L293N

- **OUT1 і OUT2** – роз'єми для підключення першого щіткового двигуна або першої обмотки крокового двигуна;
- **OUT3 і OUT4** – роз'єми для підключення другого щіткового двигуна або другої обмотки крокового двигуна;
- **VSS** – вхід для живлення двигунів (максимальний рівень +35V);
- **GND** – загальний провід (обов'язково з'єднати з аналогічним входом Arduino);
- **Vs** – вхід для живлення логіки +5V. Через нього безпосередньо живиться сама мікросхема L298N. Є ще другий спосіб живлення, при якому 5V для L298N береться від вбудованого в модуль стабілізатора напруги. В такому разі на роз'єм подається тільки живлення для двигунів (Vss), контакт Vs залишається не підключеним, а на платі встановлюється перемичка живлення від стабілізатора, який обмежить напругу живлення двигунів до прийнятних 5V;
- **IN1, IN2** – контакти управління першим щітковим двигуном або першої обмоткою крокового двигуна;

- **IN3, IN4** – контакти управління другим щітковим двигуном або другою обмоткою крокового двигуна;
- **ENA, ENB** – контакти для активації/деактивації першого і другого двигунів або відповідних обмоток КД. Подача логічної одиниці на ці контакти дозволяє обертання двигунів, а логічний нуль – забороняє. Для зміни швидкості обертання щіткових моторів на ці контакти подається ШІМ-сигнал. Для роботи з кроковим двигуном, як правило, на ці контакти ставлять перемички, що забезпечують постійну підтяжку до +5V.

#### **Технічні характеристики:**

- Напруга живлення логіки: 5V;
- Споживаний логікою струм: 36mA;
- Напруга живлення двигунів: від 5V до 35V;
- Робочий струм драйвера: 2A;
- Піковий струм драйвера 3A;
- Максимальна потужність: 20 Вт (при температурі 75° C);
- Розміри модуля: 43.5 x 43.2 x 29.4 мм.

Працювати з модулем L298N досить просто і комфортно. Тут не доведеться вивчати тонни літератури з описом заморочених протоколів. Все зводиться до простого перемикання логічних рівнів на виводах Arduino та до генерації ШІМ, якщо потрібно керувати швидкістю обертання.

Так як драйвер здатний керувати двома типами двигунів (щітковим або кроковим), то і робота з ним будується по-різному. Розглянемо підключення звичайних щіткових двигунів постійного струму і управління ними з допомогою плати Arduino (модифікація не важлива). На рис. 3.2 показана відповідна схема підключення.

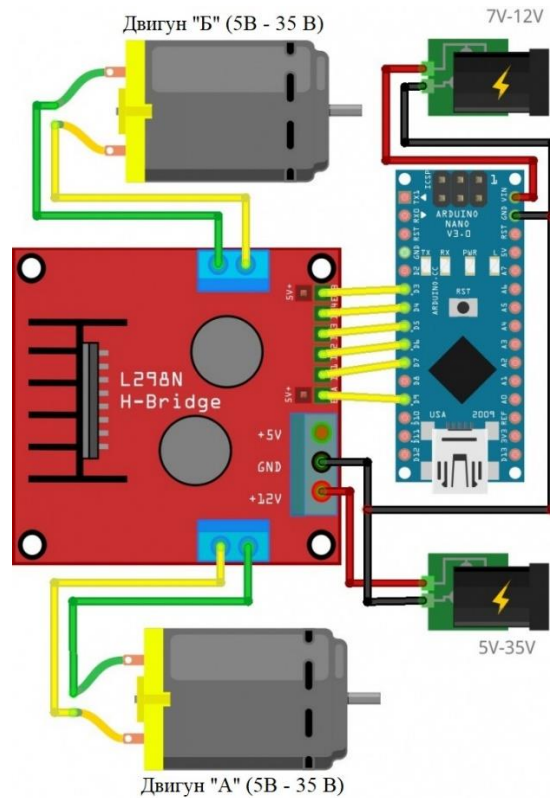


Рисунок 3.2 - Підключення щіткових двигунів постійного струму до драйвера L293N

### 3.3 Індуктивний датчик положення LJ8A3-1-Z-AX NPN NC

Призначений для безконтактного контролю стану предметів виготовлених з електропровідних матеріалів. Без фізичного контакту і незалежно від форми, індукційні безконтактні датчики виявляють предмети з кольорових і чорних металів, що знаходяться в межах активної зони, і видають відповідний керуючий сигнал. В даному проекті передбачається використання його в якості обмежувача кінцевих положень системи, та для точного позиціонування на необхідному рівні (режим роботи - лічильник) [26,27].

#### Характерні особливості безконтактних перемикачів

- Не вимагають обслуговування і мають стійкість до зносу;
- Не потребують фізичного контакту;
- Відсутність фізичного контакту призводить до відсутності биття;
- Висока частота спрацьовувань;
- Установка в будь-якій позиції;

- Термін служби не залежить від частоти роботи;
- Нечутливі до вібрацій;
- Не накопичують бруд;
- Водостійкі;
- Висока стійкість до хімічних реагентів.

#### **Технічні характеристики**

- Робоча напруга 6...30 В;
- Максимальний струм виходу 100 мА;
- Відстані спрацьовування 4 мм;
- Гістерезис до 10% від дистанції;
- Довжина проводу до 2 м.

### **3.4 Модуль бездротового зв'язку Bluetooth HC05**

Контролер Arduino не підтримує бездротовий зв'язок, в нашому випадку це просто необхідно для виконання задачі віддаленого управління системою. В цілому, для виконання даного завдання можна використати безліч модулів для організації різних технологій бездротового зв'язку: модулі WiFi, GSM/GPRS, IR, Bluetooth, радіомодули для роботи в різних частотних діапазонах [28].

Технологія Bluetooth використовується для передачі даних між двома пристроями, які знаходяться в безпосередній близькості один з одним, причому обов'язкова пряма видимість. Технологія Bluetooth забезпечує хорошу стійкість до широкосмугових перешкод, що дозволяє безлічі пристроїв, що перебувають в одному місці, одночасно спілкуватися між собою, не заважаючи один одному. Дуже широко дана технологія використовується в телефонах, планшетах, ноутбуках.

Одне з кращих рішень для організації двостороннього зв'язку по Bluetooth нашого Arduino-пристрою з планшетом, ноутбуком або іншим Bluetooth-пристроєм Bluetooth – модуль HC-05, який може працювати як

master (здійснювати пошук Bluetooth-пристроїв і ініціювати встановлення зв'язку), так і slave (ведений пристрій).

### **Технічні характеристики модуля HC05**

- Чіп Bluetooth: HC-05(BC417143);
- Діапазон частот радіозв'язку: 2,4–2,48 ГГц;
- Потужність передачі: 0,25–2,5 мВт;
- Чутливість: -80 дВм;
- Напруга живлення: 3,3–5 В;
- Споживаний струм: 50 мА;
- Радіус дії: до 10 метрів;
- Інтерфейс: послідовний порт;
- Режими: master, slave;
- Температура зберігання: -40...85 °С;
- Робочий діапазон температур: -25...75 °С;
- Габарити: 27 x 13 x 2,2 мм.

### **Підключення**

Датчик має 6 виводів стандартною відстанню між шпильками – 2,54 мм:

- VCC — (живлення 3,6В – 6В);
- GND — (земля);
- TXD, RXD — інтерфейс UART;
- STATE — індикатор стану;
- KEY / EN — контакт для входу в режим програмування.

На рис. 3.3 зображено простий спосіб підключення модуля HC05 до цифрових виводів плати Arduino. В такому випадку необхідно підключити додатково бібліотеку SoftwareSerial.h для реалізації функцій послідовного порта на цифрових пінах плати [29].

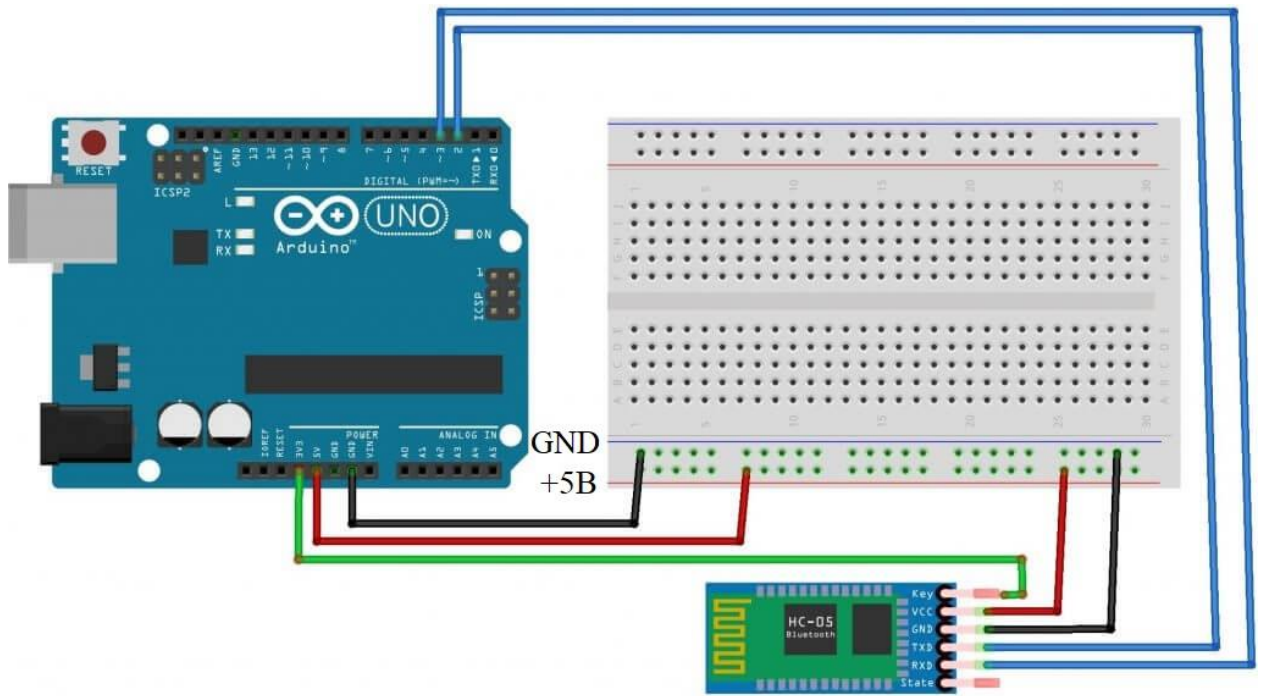


Рисунок 3.3 – Один з варіантів підключення модуля HC05

### 3.5 Модуль бездротового зв'язку Wi-Fi ESP-01

Додатково розглядається варіант підключення планшету до системи за рахунок мережі Wi-Fi.

Wi-Fi модуль ESP-01 – найпопулярніший модуль серії ESP8266. Спілкування з комп'ютером або мікроконтролером здійснюється через UART за допомогою набору AT-команд. Крім того, модуль можна використовувати як самостійний пристрій, для цього необхідно в нього завантажити свою прошивку. Програмувати і завантажувати прошивки можна через Arduino IDE версії вище 1.6.5. Для прошивки модуля знадобиться перехідник UART-USB. Модуль ESP-01 може одержати широке поширення для використання в пристроях IoT (Інтернет речей) [30,31].

#### Технічні характеристики модуля

- Wi-Fi 802.11 b/g/n;
- Режими WiFi: клієнт, точка доступу;
- Вихідна потужність - 19,5 дБ;
- Напруга живлення 1.8 В -3.6;

- Струм споживання - 220 мА;
- Портів GPIO : 4;
- Тактова частота процесора - 80 МГц;
- Оперативна пам'ять - 96 КБ;
- Розміри - 13×21 мм.

### Підключення

Розглянемо режим АТ-команд. Для цього підключимо модуль до комп'ютера через перехідник USB-UART. Призначення виводів модуля (див. рис. 3.4):

- VCC — +3.3 В;
- GND — земля;
- RX, TX — виводи UART;
- Вивод CH\_PD — Chip enable;
- GPIO0, GPIO2 — цифрові контакти;

Модулю потрібний зовнішнє живлення 3.3 В.

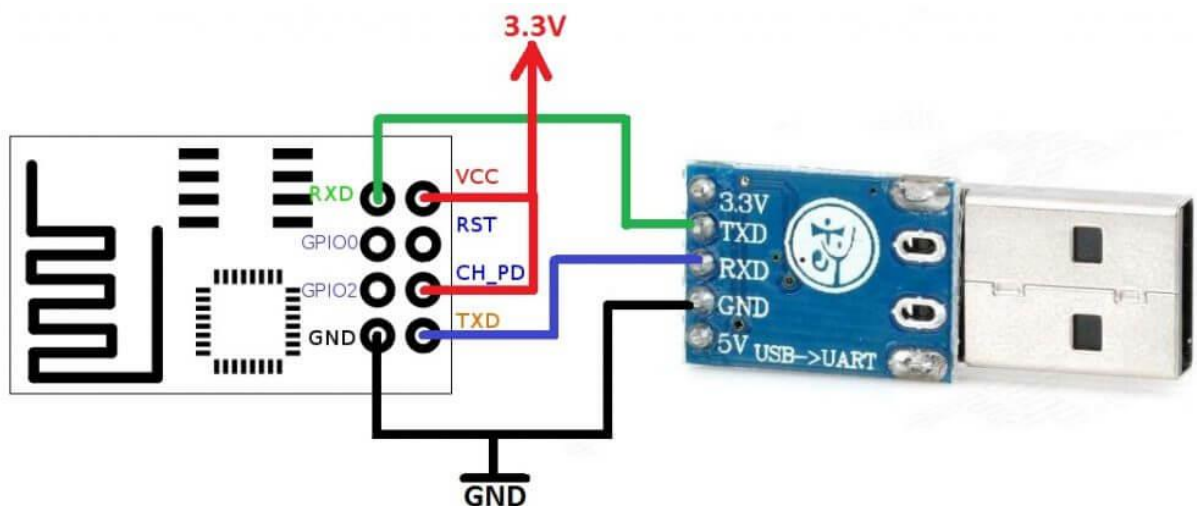


Рисунок 3.4 – Призначення виводів модулю ESP-01 та спосіб підключення через перехідник USB-UART (аналогічно для Arduino)

### 3.6 RFID модуль RC522 з карткою доступу для Arduino

RFID (англ. Radio Frequency IDentification, радіочастотна ідентифікація) - спосіб автоматичної ідентифікації об'єктів, в якому за допомогою радіосигналів зчитуються або записуються данні, що зберігаються в так званих транспондерах, або RFID-мітках.

Будь-яка RFID-система складається з пристрою, що зчитує (зчитувач, рідер або інтеррогатор) і транспондера (він же RFID-мітка, іноді також застосовується термін RFID-тег).

RFID модуль для карт Mifare на RC522 використовується для читання / запису RFID-карт фірми Mifare. Модуль може бути використаний в системах контролю доступу, автоматичної ідентифікації, робототехніки тощо Практичне застосування: автоматичний замок для відкривання дверей, ідентифікатор доступу Windows блокування / розблокування за допомогою RFID і т.п [30,31].

#### Технічні характеристики модуля

- Струм: 13-26 mA / DC 3.3V;
- Струм очікування: 10-13 mA / DC 3.3V;
- Струм спокою <80 uA;
- Піковий струм: <30 mA;
- Робоча частота: 13.56 MHz;
- Карти: S50, S70, UltraLight, Pro, Desfire;
- Фізичні характеристики: Розміри: 40мм x 60мм;
- Робоча температура: -20 + 80C;
- Температура зберігання: -40 +85 C.



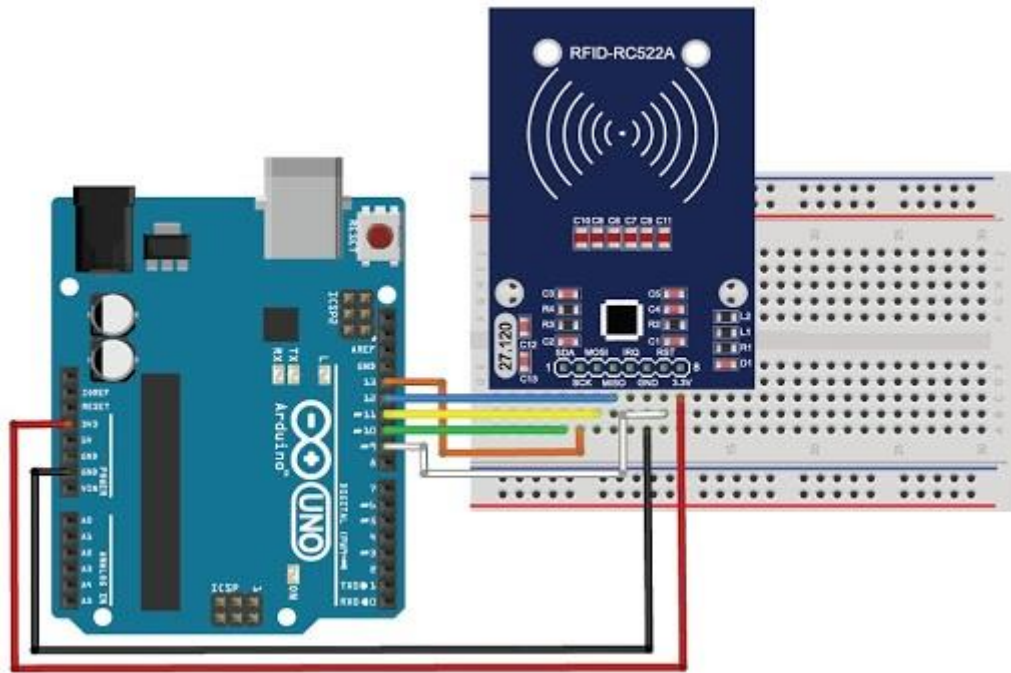


Рисунок 3.5 – Схема підключення RFID модуля RC522

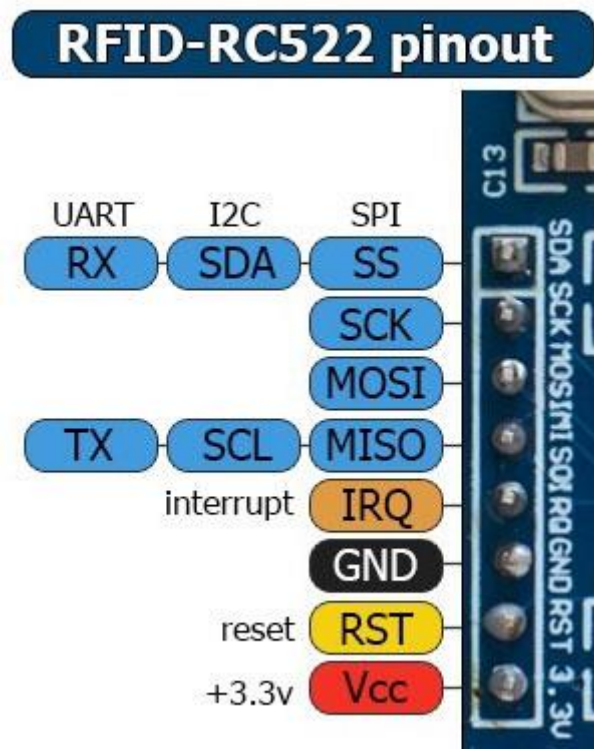


Рисунок 3.6 – Піни RFID модуля RC522

Мікросхема MFRC522 підтримує інтерфейси SPI, UART і I2C. Вибір інтерфейсу здійснюється установкою логічних рівнів на певних виводах мікросхеми. В Arduino прийнято використовувати **SPI**.

Призначення виводів інтерфейсу SPI:

- SDA - вибір веденого;
- SCK - сигнал синхронізації;
- MOSI - передача від master до slave;
- MISO - передача від slave до master;
- IRQ - висновок переривання;
- GND - земля;
- RST - висновок для скидання;
- Vcc - живлення 3.3 В.

На платах Arduino є роз'єм ICSP, який можна використовувати для роботи з інтерфейсом SPI.

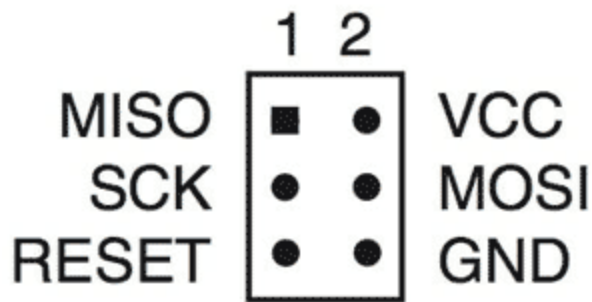


Рисунок 3.7 – Роз'єм ICSP Arduino

### 3.7 Система живлення пристроїв

Щоб забезпечити всі вузли проекту живленням необхідно передбачити основний блок живлення потужністю мінімум 60W. Потенційно, вибір падає на негерметичний БЖ в перфорованому металевому корпусі зі стабілізованою вихідною напругою 12V постійного струму. Підключення до мережі 220V і підключення споживачів перетвореної напруги відбувається за допомогою клем гвинтового кріплення. Даний блок живлення відрізняється більш компактними розмірами в лінійці аналогічних за потужністю апаратів, що дозволяє використовувати його навіть в дуже обмеженому просторі. Блок живлення захищений від короткого замикання і перевантаження, що сприяє виключенню виходу з ладу [32].

БЖ без активної системи охолодження, необхідно передбачати наявність природної циркуляції повітря в місцях установки БП, або систему примусового обдуву.

Для реалізації живлення модулів, розрахованих на напругу 5В передбачено встановлення понижуючого конвертера постійного струму, побудованого на базі LM2596S. Діапазон напруг на вході 4.5 - 40В, на виході 3 - 35В (обов'язкова умова: вхідна напруга повинна бути не менш ніж на 1.5 вище ніж вихідна).

### **Характеристики:**

- Випрямляч: не синхронне випрямлення;
- вхідна напруга: 4.5 - 40В;
- вихідна напруга: 3 - 35В (регулюється підлаштувальним резистором, розташованим на платі);
- вихідний струм: 2А (3А максимум при активному охолодженні)
- ефективність перетворення: 92% (чим вище вихідна напруга, тим вище ефективність);
- споживання без навантаження: 6мА;
- мінімально можлива різниця між вхідним і вихідним напругою 1.5В;
- швидкість динамічного відповіді: 5% 200мкс;
- захист від короткого замикання: обмеження струму;
- захист від переполюсовки на вході: немає;

Попередньо схема з'єднань має вигляд, зображений на рис. 3.8.

Примітка: попередня схема з'єднань може містити не точності та не включати певні елементи схеми, достовірна інформація розміщена на плакатах «схема електрична принципова» та «схема електрична з'єднань».

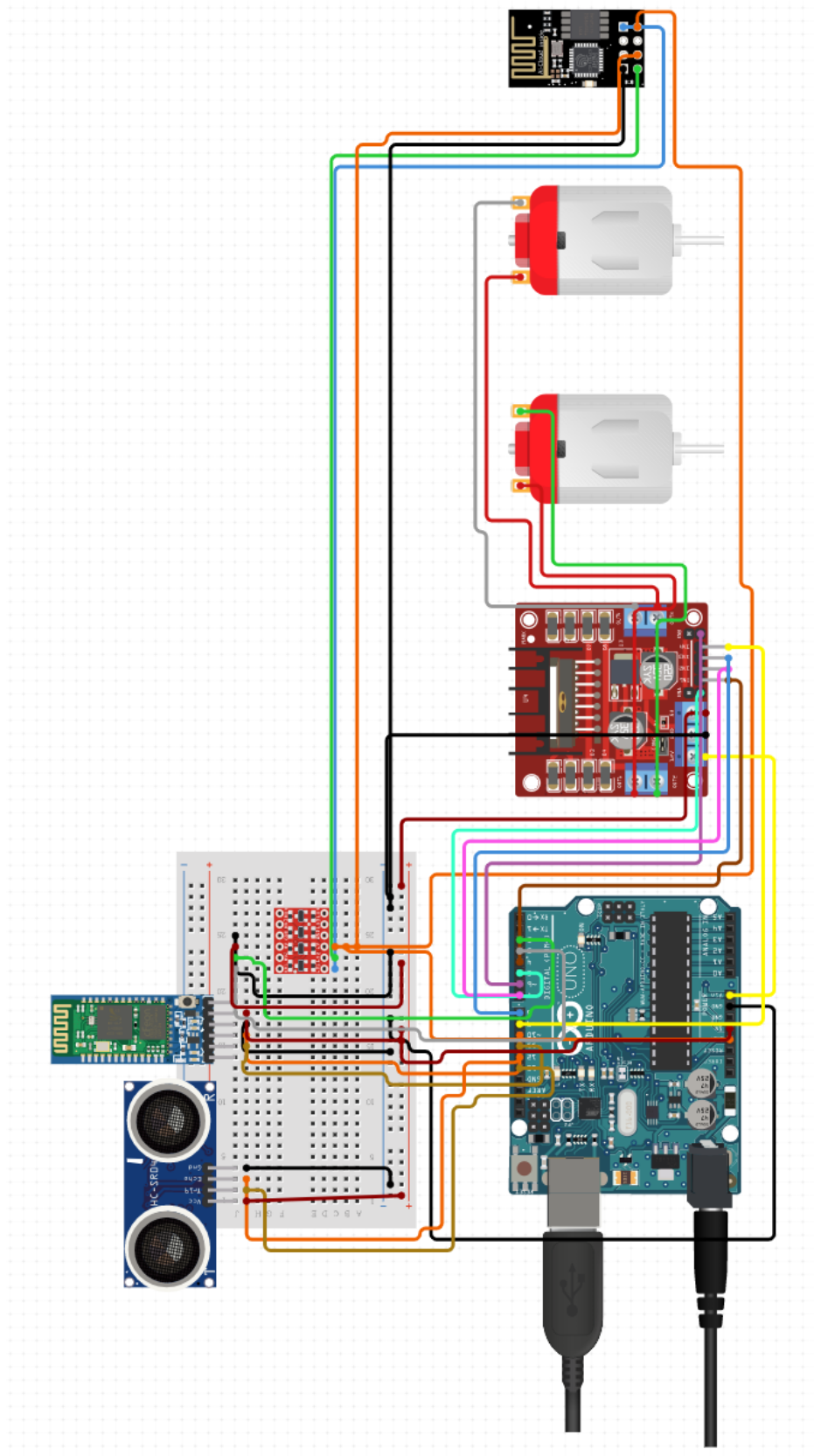


Рисунок 3.8 – Попередній вигляд схеми з'єднань проекту

### **Висновки до розділу**

1. Обрано набір основних елементів для забезпечення необхідного функціоналу проєкту.
2. Можливе дистанційне керування за допомогою планшету через Bluetooth та/або Wi-Fi.
3. Передбачається попередня авторизація «клієнтів» паркінгу.
4. Проєкт потребує декількох номіналів для живлення вузлів, в тому числі 12 В та 5 В постійного струму.
5. Зібрано попередню схему з'єднань обраного обладнання.

## РОЗДІЛ 4. ВИБІР І РОЗРОБКА АПАРАТНОГО ЗАБЕЗПЕЧЕННЯ

Інженерія управління еволюціонувала з плином часу. У минулому саме люди були основним методом управління системою. Зовсім недавно електротехніку було використано для керування процесами, і раннє електричне управління базувалося на електромеханічному реле. Ці реле дозволяють вмикати і вимикати живлення без механічного вимикача, і, зазвичай, використовувалися для прийняття простих логічних керуючих рішень. Розвиток недорогого «комп'ютера» приніс найостаннішу революцію -програмований логічний контролер (ПЛК). Поява ПЛК в реальних проектах почалася в 1970-х роках і швидко зайняла перші місця серед найпопулярніших способів управління виробництвом. ПЛК набирають популярність на заводах і, ймовірно, будуть залишатися домінуючими протягом ще тривалого часу [33].

Велика частина цього відбувається через переваги, які вони пропонують, одні з них:

- Економічність для управління складними системами.
- Гнучкість і можливість бути повторно застосованим для швидкого і легкого управління іншими системами.
- Обчислювальні здібності дозволяють реалізувати більш складний алгоритм керування.
- Засоби усунення несправностей полегшують програмування та скорочують час простою.
- Надійність електронних компонентів забезпечують працездатність автоматизованих комплексів протягом багатьох років до відмови.

### 4.1 Використання ПЛК на базі Arduino в практиці

Як вже згадувалося раніше, важливі модулі PLC схожі на ПК (персональний комп'ютер) і навіть більше – схожі на одноплатні комп'ютери, такі як Arduino. Таким чином, внутрішньо на певному рівні робота ПЛК і Arduino однакова, і ми можемо використовувати Arduino для розробки ПЛК

(програмованого логічного контролера). ПЛК Arduino вже представлені на ринку і доступні в порівнянні зі звичайними ПЛК.

Таким чином, ПЛК Arduino стає популярним в наші дні, і його додатки будуть рости в майбутньому. Але є певні відмінності між ПЛК Arduino і звичайним ПЛК, і деякі з них згадані далі.

По-перше, Arduino необхідні зовнішні компоненти для роботи в якості ПЛК, в самому ПЛК не потрібні. Важливою перевагою Arduino є його низька вартість. Для Arduino потрібно навчитися основам програмування, щоб переписати програму, у випадку з ПЛК необхідна тільки базова техніка для перепрограмування ПЛК.

У Arduino задовільна продуктивність, в той час як у ПЛК продуктивність для додатків промислової автоматики значно вище завдяки їх цільовій спрямованості. Arduino не може працювати в суворих умовах, PLC може. Але плати Arduino легкі і компактні, ПЛК важкі і громіздкі. І Arduino легко замінити або відремонтувати, з ПЛК це зробити складніше [34].

Переваги ПЛК на основі Arduino:

- Можна купити за низькою ціною.
- Можуть бути запрограмовані з використанням програмного забезпечення Arduino IDE.

- Висока сумісність.

- Великі можливості для налаштування.

- Легко замінити в порівнянні зі звичайним ПЛК.

Недоліки ПЛК на основі Arduino:

- Дуже мало варіантів доступних для вибору.

- Не підходить для застосування у масштабних проектах.

- Чутливіші до завад та інших факторів в порівнянні зі звичайним ПЛК.

- Потребують більше уваги та обслуговування.

Менш професійні, що тягне за собою очевидні наслідки.

Багато конфігурацій ПЛК доступні від одного постачальника. Але, в кожному з них є загальні компоненти і поняття. Найбільш важливими компонентами є:

- Джерело живлення – може бути вбудований в ПЛК або зовнішнім. В загальному, рівні напруги, необхідні ПЛК (з блоком живлення і без нього), складають – 24 В постійного струму, 120 В змінного струму, 220 В змінного струму.
- CPU (центральний процесор).
- Введення/виведення (вхід/вихід) – необхідно передбачити кілька вхідних / вихідних клем для реалізації функцій керування. ініціалізації алгоритмів тощо.
- Індикаторні лампи (світлодіоди) – наочно демонструють стан ПЛК, дозволяють діагностувати режим роботи самого контролера, стан входів/виходів, живлення, запуск програми, індикація несправності тощо.

## 4.2 Входи і виходи

Входи і виходи ПЛК необхідні для моніторингу та управління процесами в керованій системі. Входи і виходи можна розділити на два основних типи: логічні і безперервні. Розглянемо приклад на основі електричної лампочки. Процес включання та вимикання лампочки – це логічний контроль. Якщо світло можна приглушити до різних рівнів (димірувати), то спосіб керування безперервний. Безперервні (наприклад, ШІМ чи аналогові сигнали) значення здаються більш інтуїтивними, але «логічні» значення кращі тим, що забезпечують більшу визначеність і спрощують управління в цілому.

В результаті більшість керуючих додатків (і ПЛК) використовують логічні входи і виходи. В проєктованій парковці мають місце обидва типи виходів з контролера Arduino.

Широтно-імпульсна модуляція (ШІМ) це спосіб управління потужністю на навантаженні за допомогою зміни шпаруватості імпульсів при постійній амплітуді і частоті імпульсів.



Можна виділити дві основні області застосування широтно-імпульсної модуляції:

- У вторинних джерелах живлення, різних регуляторах потужності, регулятора яскравості джерел світла, швидкості обертання колекторних двигунів і т.п. у цих випадках застосування ШІМ дозволяє значно збільшити ККД системи і спростити її реалізацію.

- Для отримання аналогового сигналу за допомогою цифрового виходу мікроконтролера. Своєрідний цифро-аналоговий перетворювач (ЦАП). Дуже простий в реалізації, вимагає мінімуму зовнішніх компонентів. Часто достатньо одного RC ланцюжка.

Виходи на виконавчі механізми дозволяють ПЛК викликати будь-які події в процесі керування. Нижче наведено короткий список популярних засобів в порядку відносної частоти застосування.

- Електромагнітні клапани – логічні виходи, які можуть перемикаєти гідравлічний або пневматичний потік.

- Пускачі двигунів – двигуни часто споживають великі струми під час запуску, тому вони вимагають окремих пускатрів чи відповідних драйверів.

- Серводвигуни – безперервний вихід від ПЛК може управляти змінною швидкістю або положенням приводу. Виходи з ПЛК часто є релейними, але також можуть бути твердотільними, наприклад транзистори для виходів постійного струму або сімістори для виходів змінного струму. Аналогові виходи вимагають наявності спеціальних додатків з цифроаналоговими перетворювачами, які можуть бути як вбудовані в ПЛК/контролер (як в нашому випадку), так і використовуватися як окремий пристрій.

Вхідні сигнали надходять від датчиків, які перетворюють фізичні явища в електричні сигнали. Типові приклади датчиків перераховані нижче у відносному порядку.

- Безконтактні перемикачі – використовують індуктивність, ємність або світло (освітленість) для виявлення об'єкта.

- Перемикачі – механічні пристрої, які відкривають або закривають електричні контакти для отримання логічного сигналу.
- Потенціометр – вимірює кутові положення безперервно, використовуючи опір.
- LVDT (лінійний змінний диференціальний трансформатор) – вимірює лінійне зміщення безперервно за допомогою магнітного зв'язку.

Входи для ПЛК бувають декількох основних різновидів, найпростіші – це входи змінного і постійного струму. Також популярні Sourcing та Sinking режими роботи входів. Ці методи визначають те, що пристрій не споживає ніякої енергії. Замість цього, виходи ПЛК/контролера тільки вмикають або вмикають струм, як простий перемикач.

Sinking – коли вихід активний, він дозволяє струму текти до загальної землі. Найчастіше використовується, коли підключені пристрої використовують різний тип напруги живлення.

Sourcing – коли вихід активний, струм тече від джерела живлення, через вихідний пристрій і на землю. Цей метод найкраще використовувати, коли всі пристрої використовують одну напругу живлення.

### **4.3 Алгоритм відпрацювання завдання проєкту**

На основі вказаного вище – розроблювана автоматизована система паркування автомобілів складається з 3 основних компонентів: Arduino Mega 2560 R3, Bluetooth HC-05 та RFID модулю. Основною частиною цієї системи є мікроконтролер Arduino Mega. Програмний код створено з використанням фірмового середовища програмування ArduinoIDE.

В якості дисплею та пульта керування використовується планшетний ПК з вбудованим модулем Bluetooth. Основна його задача – приймання інформації про стан системи, відображення актуальної інформації на дисплеї, відправка команд від користувача в систему. Не менш важливою частиною – є дружній та зручний інтерфейс написаного ПЗ для планшетного ПК.

По приїзду авто, оператор (користувач) надішле інструкцію через планшетний ПК до arduino про необхідність запуску процесу паркування авто за допомогою Bluetooth. Інший спосіб – користувач авторизується за допомогою RFID-мітки біля «пропускного пункту», система автоматично приймає подальші рішення про завантаження чи відвантаження авто користувача.

Після здійснення передачі даних, двигуни постійного струму підйомного механізму (ліфту) починають переміщення конструкції завантажувача палети до необхідного рівня. Після спрацювання одного з індуктивних датчиків перешкод, починається процес відвантаження авто до паркомісця. В залежності від розміщення вільного паркомісця (ліворуч чи праворуч по відношенню до автомобіля), запускається один з двох двигунів механізму відвантаження палети з авто, котрі за допомогою передачі «шестерня – зубчаста рейка» автомобіль на палеті поступово займає відведене для нього місце на певному ярусі парковки. Після замикання електричних контактів на рейці, які свідчать про повне висунання механізму, двигун зупиняється. Механізм підйому переміщується вниз, доки палета з авто повністю не розміститься на рейках-тримачах основної конструкції парковки, після чого механізм відвантаження авто повертається в вихідне положення, механізм підйому спускається в місце прийому (нульовий поверх/ярус).

#### 4.4 Блок-схема алгоритмів

На рис. 4.1 зображено основний алгоритм, необхідний для паркування авто та його подальшого відвантаження з паркомісця. Різниця між процесом паркування та відвантаження полягає в спрацюванні різних індуктивних давачів перешкод, розрахованих на розміщення платформи на двох рівнях.

При паркуванні авто – платформа підіймається не значно вище напрямних для палети, при відвантаженні – навпаки, злегка нижче встановленої палети з автомобілем.

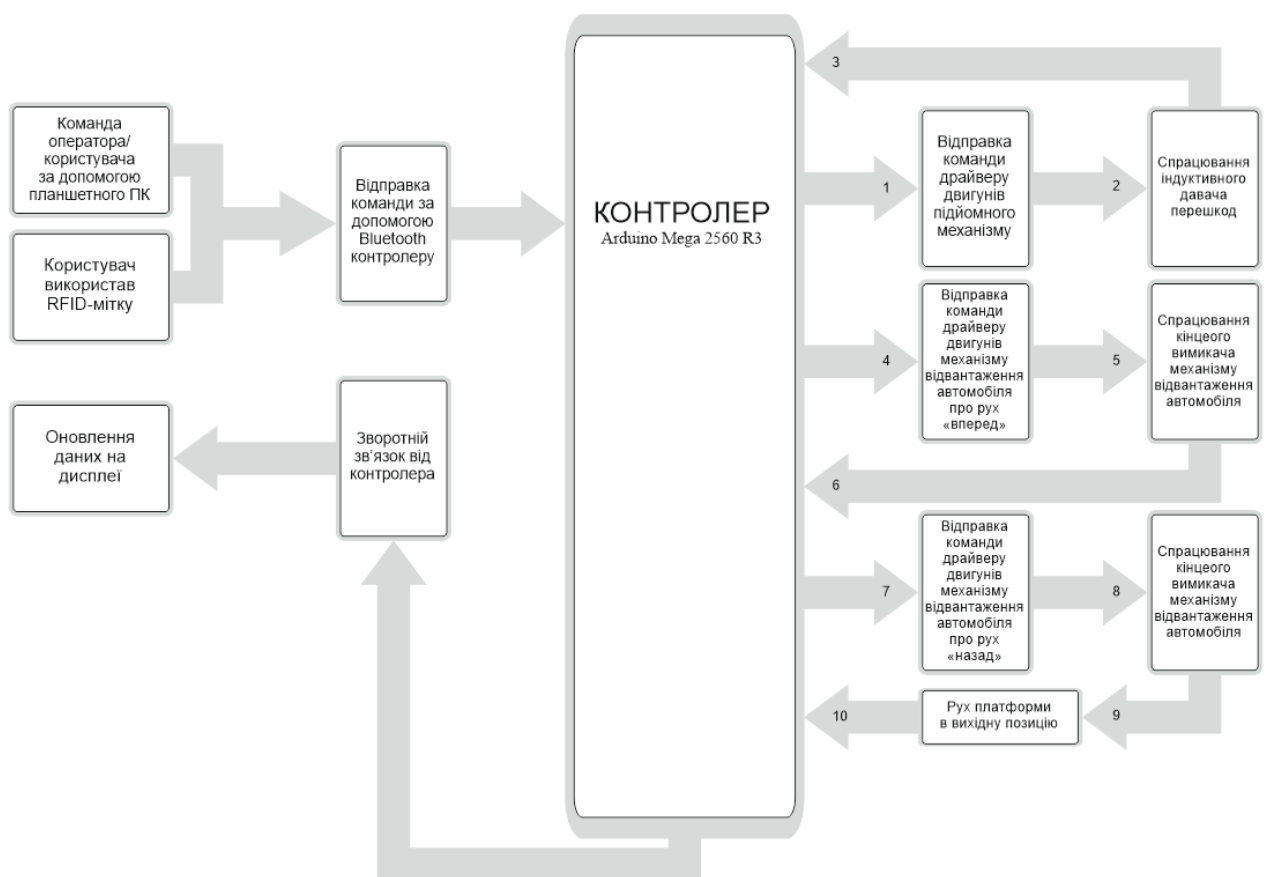


Рисунок 4.1 – Алгоритм роботи автоматизованої парковки

#### 4.5 Інтерфейс користувача

Інтерфейс програми керування за допомогою планшету/окремої панелі, на базі операційної системи Android виконано в мінімалістичному, легкому дизайні, без банерів та відволікаючих увагу користувача елементів.

Зовнішній вигляд головного меню зображено на рис. 4.2. Інтерфейс передбачає наступні елементи:

- Привітання користувача, який має RFID-ідентифікатор;
- Напис-лічильник вільних місць;
- Актуальна дата та час;
- Актуальна вартість послуг із вказанням термінів паркування;
- 6 подіумів з нумерацією, відповідно зображено кожна комірка парковки;
- Зображення автомобіля на паркомісцях, які вже зайняті;
- Праворуч панель керування, яка дозволяє як забрати автомобіль з парковки, так і залишити авто.
- Панель керування візуалізує доступність комірки для керування користувачем;
- Нижче розміщено логотип та слоган компанії.

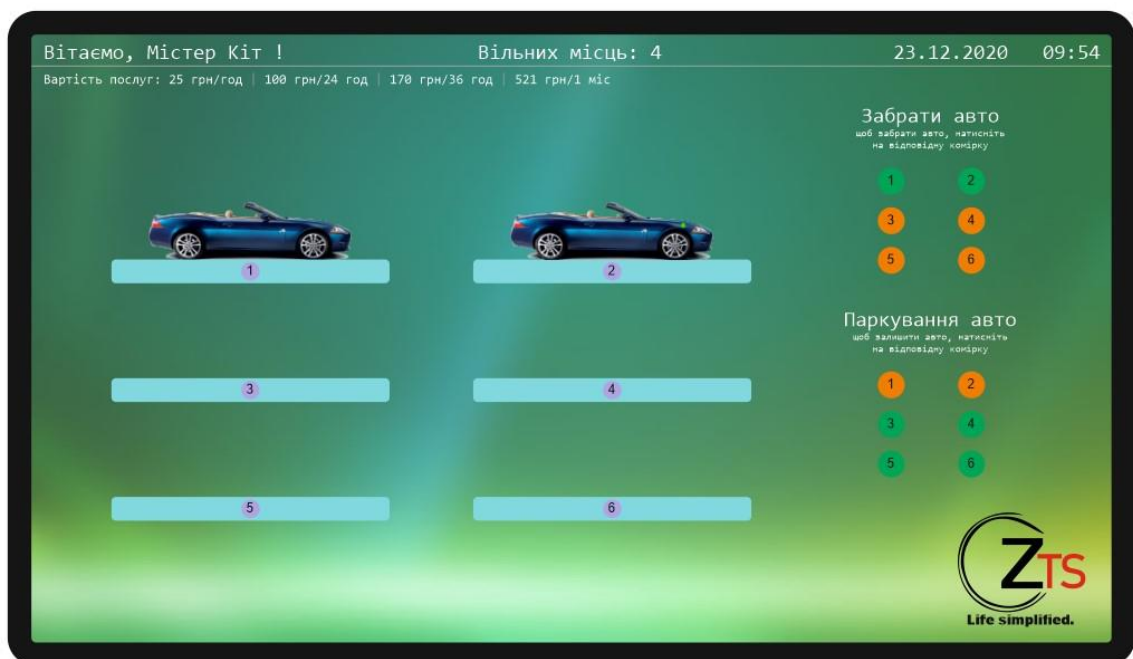


Рисунок 4.2 – Головне меню ПЗ на ОС Android

Алгоритм проходження процесу авторизації та паркування/відвантаження автомобіля представлено на рис. 4.3. Користувач обов'язково має пройти попередню авторизацію за допомогою RFID-мітки.

Практично, абстрагуючись від контексту дипломного проекту, реєстрацію можна пройти в процесі паркування, за допомогою телефонного номеру та отримати пропуск-мітку відразу.

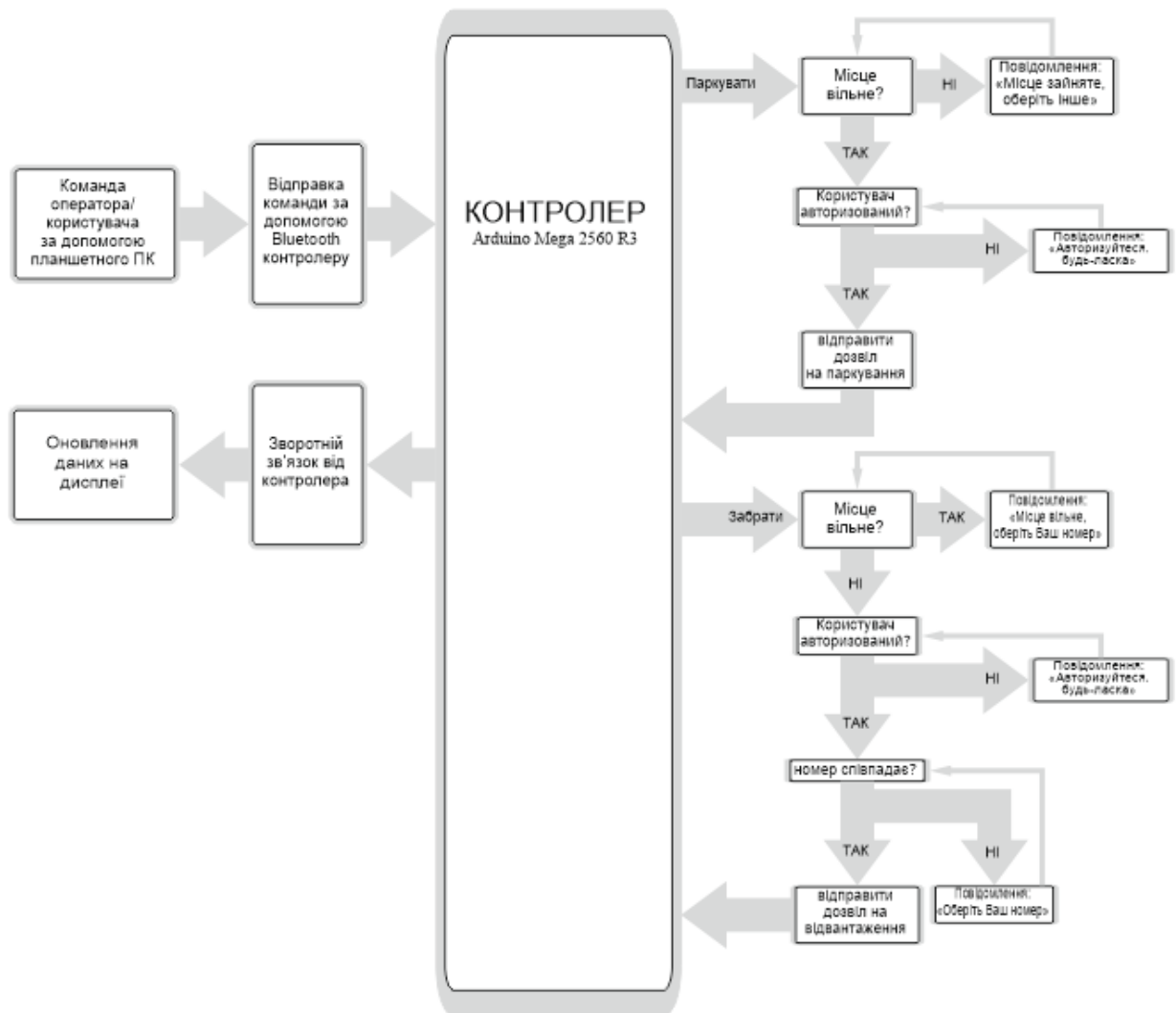


Рисунок 4.2 – Алгоритм проходження процесу авторизації та паркування/відвантаження автомобіля

### **Висновки до розділу**

1. Платформа Arduino завдяки своїй низькій вартості та, не зважаючи на не високу продуктивність, знаходить своє застосування в досить різноманітних за цільовим призначенням проектах.
2. В цьому розділі описано основні алгоритми, необхідні для запуску працездатної системи, котрі зв'язують як пульт керування користувача (включно з графічною складовою), так і роботу двигунів на основі давачів перешкод, які використовуються в тому числі як лічильники (для визначення цільового рівня паркінгу).
3. Інтерфейс пов'язує в собі легкість та функціонал, не перевантажує користувача зайвою інформацією та дає змогу без використання додаткових сторінок меню виконувати як авторизацію, так і паркування/відвантаження автомобіля.

## РОЗДІЛ 5. РОЗРОБКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

### 5.1 Управління ДПС за допомогою ARDUINO

Бібліотека підходить для більшості драйверів, побудованих за схемою Н-міст, на два мотора вони зазвичай мають 4 входи (по 2 на кожен). Також на сайті розробника є окремий детальний урок по роботі з колекторними двигунами, використовуючи описану нижче бібліотеку [35].

Актуальна версія бібліотеки GyverMotor v3.1 для зручного управління двигунами через драйвер повного моста для Arduino має наступні можливості:

- Контроль швидкості і напрямку обертання;
- Робота з 10 бітовим ШІМом;
- програмний deadtime;
- негативні швидкості;
- Підтримка двох типів драйверів та реле;
- Плавний пуск і зміна швидкості;
- Підтримувані платформи : все Arduino (використовуються стандартні Wiring-функції);
- У версії 2.2 додана підтримка плат на базі ESP.

#### 5.1.1 Ініціалізація бібліотеки

Бібліотека підтримує три типи драйверів:

- **DRIVER2WIRE** - двопровідний драйвер (напрямок + ШІМ);
- **DRIVER2WIRE\_NO\_INVERT** - двопровідний драйвер, в якому при зміні напрямку не потрібна інверсія ШІМ;
- **DRIVER3WIRE** - трьохпровідний драйвер (два Піна напрямки + ШІМ);
- **RELAY2WIRE** - реле як драйвер (два Піна напрямки);

Ініціалізація відбувається наступним чином:

- GMotor motor ( DRIVER2WIRE, dig\_pin, PWM\_pin, level );
- GMotor motor ( DRIVER3WIRE, dig\_pin\_A, dig\_pin\_B, PWM\_pin, level );



— GMotor motor ( RELAY2WIRE, dig\_pin\_A, dig\_pin\_B, level ),

де:

- \* dig\_pin, dig\_pin\_A, dig\_pin\_B - будь-який цифровий пін МК;
- \* PWM\_pin - будь-який ШІМ пін МК;
- \* level - LOW / HIGH - рівень драйвера. Якщо при збільшенні швидкості мотор навпаки гальмує – необхідно змінити рівень.

### 5.1.2 Налаштування бібліотеки

— setDeadtime ( us ) - установка програмного deadtime на перемикання напрямку, в мікросекундах. За замовчуванням встановлено 0: deadtime відключений;

— setDirection ( dir ) - ГЛОБАЛЬНА зміна напрямку обертання мотора наприклад щоб FORWARD збігався з напрямком руху "вперед" у машини;

— dir - REVERSE або NORMAL (Стандарт.);

— setLevel ( level ) - зміна рівня драйвера (аналогічно при ініціалізації). Якщо при збільшенні швидкості мотор навпаки гальмує - змieni рівень;

— Level - LOW або HIGH;

— setMinDuty ( duty ) - мінімальний сигнал (по модулю), який буде поданий на мотор. Автоматично стискає діапазон регулювання мотора: наприклад minDuty поставили 50, і при сигналі 1 буде на мотор буде подано ~ 51, максимум залишиться колишнім (діапазон сигналу переведе в 50 .. 255 всередині бібліотеки);

### 5.1.3 Режим роботи двигунів

У двигуна є 5 режимів роботи, встановлюються за допомогою setMode ( mode ), де mode:

- FORWARD – вперед;
- BACKWARD – назад;
- STOP - холостий (мотор відключений);
- AUTO - віддати управління функції setSpeed ();

#### 5.1.4 Управління швидкістю і напрямком

Швидкість встановлюється за допомогою функції `setSpeed (speed )`. Є два варіанти управління швидкістю:

— Напрямок в ручному режимі, швидкість 0..255 (0..1023). У цьому випадку швидкість повинна бути позитивною (`setSpeed (0. .255)`), напрямок обертання задається за допомогою `setMode ()`. При напрямку FORWARD частота обертів зростає в міру збільшення `setSpeed (speed)`, При направленні BACKWARD частота обертів зростає в протилежну сторону в міру збільшення `setSpeed (speed)`;

— Напрямок в автоматичному режимі, швидкість -255..255 (-1023..1023 для 10 біт). В цьому випадку встановлюємо `setMode ( AUTO )` і подаємо швидкість у всьому діапазоні. При значенні -255 двигун крутиться з максимальною швидкістю в напрямку "назад", при 255 - в напрямку "вперед". При значенні 0 (нуль) режим автоматично перемикається на STOP . Якщо заданий параметр `minDuty` (за допомогою `setMinDuty ()` ), то режим STOP буде автоматично активований при значенні швидкості в межах  $(-\text{minDuty}, \text{minDuty})$ . Приклад: у нас є двигун, експериментально встановлено, що він починає обертатися при величині швидкості більше 50 (і, відповідно, менше -50), при меншому значенні він просто "пищить" і стоїть на місці. Якщо ми виставимо `setMinDuty (50)`, то в діапазоні -50..50 двигун автоматично буде знаходитися в режимі STOP і не буде пищати при "шумі" заданого сигналу.

#### 5.1.5 Роздільна здатність ШІМ

Функція `setSpeed (speed)` може працювати з ШІМ будь-якого розрішення, для цього потрібно:

— Перевести таймер в режим потрібного розрішення. 8 бітні таймери можна запустити лише на зниженому розрішенні, а 16 бітний - наприклад на 10 бітах (виводи D9 і D10).

— Налаштувати потрібний об'єкт класу `GMotor` на потрібне розрішення за допомогою `setResolution (bit)`, де `bit` - розрішення ШІМ в бітах (за

замовчуванням 8). Також не забути підключити пін драйвера ШІМ до виводу налаштованого в першому пункті таймера.

*Кілька моторів можуть працювати на одній платі з різним розрішенням, тобто, наприклад, 2 двигуни на таймері 1 з розрішенням 10 біт, і ще два на таймері 2 з роздільною здатністю 8 біт.*

#### 5.1.6 Плавне керування швидкістю

У бібліотеці реалізований готовий інструмент для плавної зміни швидкості, що може забезпечити плавний пуск і зупинку механізмів:

- `setSmoothSpeed (acc)` - установка швидкості зміни швидкості (тобто прискорення) двигуна;
- `smoothTick (speed)` - дана функція сама змінює швидкість двигуна, плавно наближаючи її до зазначеної `speed`. Функція працює по вбудованому таймеру на `millis ()` з періодом 50 мс, тобто рекомендується викликати `smoothTick (speed)` не рідше ніж кожні 50 мс.

\*Додаток 1, Dodatok 2 – приклади використання бібліотеки.

## 5.2 Програмування модуля Bluetooth HC05

Налаштування модуля здійснюється в режимі програмування відправкою AT-команд з послідовного порту. Підключимо модуль HC05 до плати Arduino за схемою з'єднань, описаною в одному з попередніх розділів. На контакт KEY модуля (або вивод 34 плати) подамо 3.3В. AT-команди будемо відправляти з монітора послідовного порту Arduino IDE. Завантажимо в Arduino скетч, текст якого додано в рамці нижче. Нам знадобиться Arduino-бібліотека `SoftwareSerial`. Швидкість UART модуля в режимі програмування 38400 бод, але може і відрізнятись, в цьому випадку слід підібрати експериментально [36].

```
#include <SoftwareSerial.h>

SoftwareSerial mySerial (2, 3); // вказуємо Піни rx і tx відповідно

void setup () {
  pinMode (2, INPUT);
  pinMode (3, OUTPUT);
  Serial.begin (9600);
  mySerial.begin (38400);
  Serial.println ( "start prg");
}

void loop () {
  if (mySerial.available ()) {
    char c = mySerial.read (); // читаємо з software-порту
    Serial.print (c); // пишемо в hardware-порт
  }
  if (Serial.available ()) {
    char c = Serial.read (); // читаємо з hardware-порту
    mySerial.write (c); // пишемо в software-порт
  }
}
```

Після завантаження скетчу відкриємо монітор послідовного порту Arduino IDE і почнемо відправляти АТ-команди, повний список яких можна переглянути в додатку 3.

Основні команди, необхідні для налаштування:

- АТ - тестова команда. Параметрів немає. Відповідь модуля: ОК;
- АТ + VERSION? - отримати версію прошивки модуля. Параметрів немає. Відповідь модуля: + VERSION: <Param> ОК, де <Param> - версія прошивки Bluetooth-модуля;
- АТ + RESET - скидання налаштувань. Параметрів немає. Відповідь модуля: ОК;
- АТ + ADDR? - отримати адресу модуля. Параметрів немає. Відповідь модуля: + ADDR: <Param>, де <Param> - адреса Bluetooth-модуля NAP: UAP: LAP;
- АТ + NAME? - отримати ім'я модуля. Параметрів немає. Відповідь модуля: + NAME: <Param>, де <Param> - ім'я Bluetooth-модуля;

— AT + NAME = <Param> - встановити нове ім'я модуля. Параметр: <Param> - ім'я Bluetooth-модуля. Відповідь модуля: + NAME: <Param> OK (або FAIL);

— AT + PSWD? - отримати пін-код доступу до Bluetooth-модуля. Параметрів немає. Відповідь модуля: + PSWD: <Param>, де <Param> - пін-код. За замовчуванням 1234;

— AT + PSWD = <Param> - встановити код доступу до Bluetooth-модуля. Параметр: <Param> - код доступу до модуля. Відповідь модуля: OK (або FAIL);

— AT + CLASS = <Param> - встановити режим роботи модуля Bluetooth-модуля. Параметр: <Param> - клас. У документації модуля не наведено можливі значення даного параметра. За замовчуванням він встановлений в 0. Якщо передбачається використовувати модуль в режимі master, значення не треба змінювати. Якщо використовувати модуль в режимі slave, при значенні параметра, рівному 0, він невидимий для пристроїв з операційною системою Android. Для видимості необхідно встановити значення параметра рівним 7936. Відповідь модуля: OK;

— AT + CLASS? - отримати клас модуля. Параметрів немає. Відповідь модуля: + CLASS: <Param>, де <Param> - клас модуля.

### 5.3 Програмування RFID модуля RC522 з карткою доступу для Arduino

Ідентифікація об'єктів проводиться за унікальним цифровому коду, який зчитується з пам'яті електронної мітки, що прикріплюється до об'єкта ідентифікації. Зчитувач містить в своєму складі передавач і антену, за допомогою яких випромінюється електромагнітне поле певної частоти [37,38].

Потрапили в зону дії пристрою, що зчитує поля радіочастотні мітки «відповідають» власним сигналом, що містить інформацію (ідентифікаційний номер товару, призначені для користувача дані і т. Д.).

Сигнал вловлюється антеною зчитувача, інформація розшифровується і передається в комп'ютер для обробки. Переважна більшість сучасних систем

контролю доступу (СКД) використовує в якості засобів доступу ідентифікатори, що працюють на частоті 125 кГц.

Це проксиміті-карти доступу (тільки читання), найпоширенішими є карти EM-Marin, а також HID, Indala. Карти цього стандарту є зручним засобом відкривання дверей і турнікетів. Але не більше. Ці карти не володіють ніякої захищеністю, легко копіюються і підробляються і, відповідно, нічого не дають для захисту об'єкта від несанкціонованого проникнення.

Справжній захист від копіювання та підробки забезпечують такі ідентифікатори, в чіпах яких реалізована криптографічний захист. Це безконтактні смарт-карти, що працюють на частоті 13,56 МГц, найбільш поширеними з них є карти Mifare®. У картах цих стандартів криптозащита організована на високому рівні, і підробка таких карт практично неможлива.

Перед написанням програми, слід розібратися про способи підключення мітки та можливими бібліотеками, котрі дозволяють спростити процес написання коду.

RFID-модуль RC522 підключається до Arduino проводами Папа-Мама в наступній послідовності:

Таблиця 5.1 – Підключення RFID-модуля RC522

<b>MFRC522</b>	<b>Arduino Uno</b>	<b>Arduino Mega</b>	<b>Arduino Nano v3</b>	<b>Arduino Leonardo / Micro</b>	<b>Arduino Pro Micro</b>
RST	9	5	D9	RESET / ICSP-5	RST
SDA (SS)	10	53	D10	10	10
MOSI	11 (ICSP-4)	51	D11	ICSP-4	16
MISO	12 (ICSP-1)	50	D12	ICSP-1	14
SCK	13 (ICSP-3)	52	D13	ICSP-3	15
3.3V	3.3V	3.3V	стаб. 3.3	стаб. 3.3	стаб. 3.3
GND	GND	GND	GND	GND	GND

5.3.1 Бібліотека MFRC522

Встановимо через менеджер бібліотек **MFRC522**.

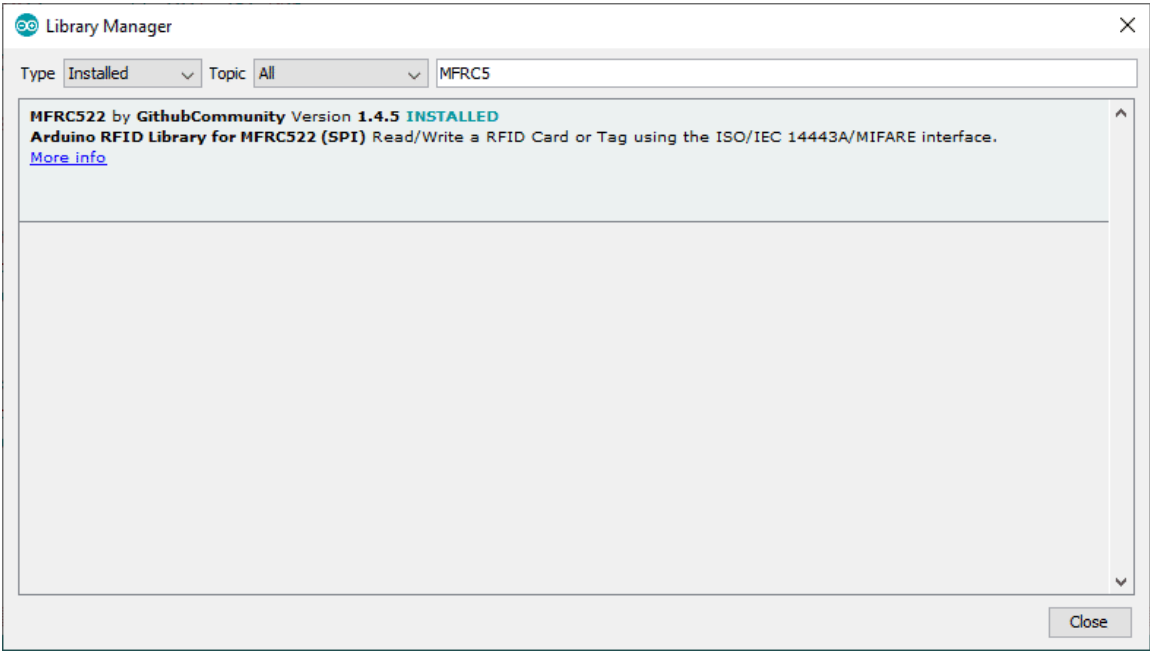


Рисунок 5.1 – Менеджер бібліотек Arduino IDE

До складу бібліотеки входять кілька прикладів, а також компонент для Fritzing. Для першого знайомства запустимо скетч **DumpInfo**.

При піднесенні картки в моніторі порту побачимо велику таблицю даних. Дочекаймося повного завантаження даних і звернемо увагу на дані в блоці 0 - там можна побачити ідентифікатор.

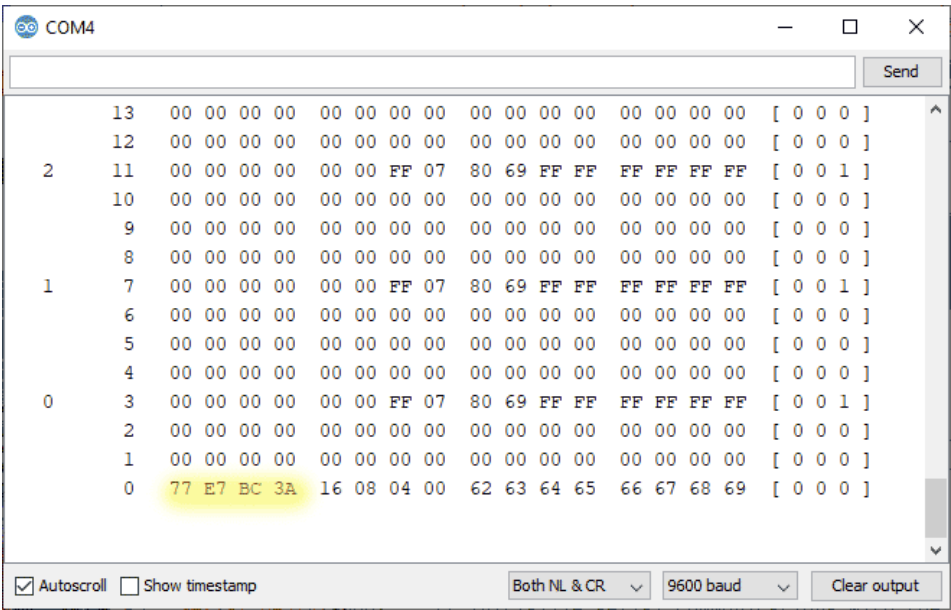


Рисунок 5.2 – Монітор послідовного порту

Але у нас немає необхідності вручну зчитувати інформацію з масиву даних, ідентифікатор можна отримати через виклик функції. На початку є окрема строчка **Card UID: 77 E7 BC 3A**.

Розглянемо спрощений приклад. Суть в наступному. Ініціалізуємо об'єкт класу **MFRC522** і перевіряємо різні умови. Якщо до рідера НЕ піднесли картку, то виходимо з функції **loop ()** , якщо прочитати картку не можемо, то також виходимо з функції. Якщо обидві умови все ж виконалися (є карта і дані з неї), то виконуємо вже потрібний код:

```

return;

#include <SPI.h>
#include <MFRC522.h>

// контакти
#define SS_PIN 10
#define RST_PIN 9

// Створення екземпляру об'єкта
MFRC522
MFRC522 mfrc522(SS_PIN, RST_PIN); //
Створення екземпляра MFRC522

void setup() {
  Serial.begin(9600);
  SPI.begin();

  // ініціалізація MFRC522
  mfrc522.PCD_Init();
  // Виводимо номер версії прошивки рідера
  mfrc522.PCD_DumpVersionToSerial();
}

void loop() {
  // Очікування
  if ( ! mfrc522.PICC_IsNewCardPresent()

return;
// Читання
if ( !mfrc522.PICC_ReadCardSerial())
return;

// вивод даних
Serial.print("UID = ");
view_data(mfrc522.uid.uidByte,
mfrc522.uid.size);
Serial.println();
Serial.print("type = ");
byte piccType =
mfrc522.PICC_GetType(mfrc522.uid.sak);

Serial.print(mfrc522.PICC_GetTypeName(p
iccType));
Serial.println();
delay(1000);
}

// перетворення в HEX
void view_data (byte *buf, byte size) {
  for (byte j = 0; j < size; j++) {
    Serial.print(buf [j].);
    Serial.print(buf [j]., HEX);
  }
}

```

Підносимо картку і брелок з комплекту (MIFARE 1 KB). Ідентифікатор картки виводиться в десятковому форматі. Кожна картка має свій ідентифікатор, який можна дізнатися після запуску скетчу бібліотеки **DumpInfo** . Запам'ятаємо ідентифікатор і напишемо свій скетч для перевірки. Якщо ідентифікатор збігається, то виводимо повідомлення (включаємо світлодіод, відкриваємо ворота і т.д.). Якщо ідентифікатор не збігається, то виводимо повідомлення про заборону.



```

#include <SPI.h>
#include <MFRC522.h>

// контакти
#define SS_PIN 10
#define RST_PIN 9

MFRC522 mfrc522(SS_PIN, RST_PIN);

// Дозволений ідентифікатор Card UID
byte uidCard[4]. = {0x77, 0xE7, 0xBC, 0x3A};

void setup() {
  Serial.begin(9600);
  SPI.begin();

  mfrc522.PCD_Init();
}

void loop() {
  // Очікування
  if ( ! mfrc522.PICC_IsNewCardPresent())
    return;

  // Читання
  if ( !mfrc522.PICC_ReadCardSerial())
    return;

  for (byte i = 0; i < 4; i++) {
    // Якщо ідентифікатор не співпадає із
    // заданим номером
    if (uidCard[i]. != mfrc522.uid.uidByte[i].)
    {
      Serial.println("У Вас немає доступу");
      return;
    }else{
      Serial.println("Доступ відкрито");
    }
  }
}

```

### 5.3.2 Запис і читання даних користувача

Розглянемо два приклади, які входять до складу бібліотеки - **rfid\_write\_personal\_data** і **rfid\_read\_personal\_data**. Вони дозволяють записати, а потім вважати дані з картки.

Спочатку завантажимо скетч для запису даних. Після прошивки підносимо картку до рідера, в моніторі порту просять ввести прізвище, яке потрібно завершити символом решітки. Після успішного запису потрібно ввести ім'я і знову з завершальним символом решітки.

Після запису – рідер зчитає дані і виведе їх в монітор порту.

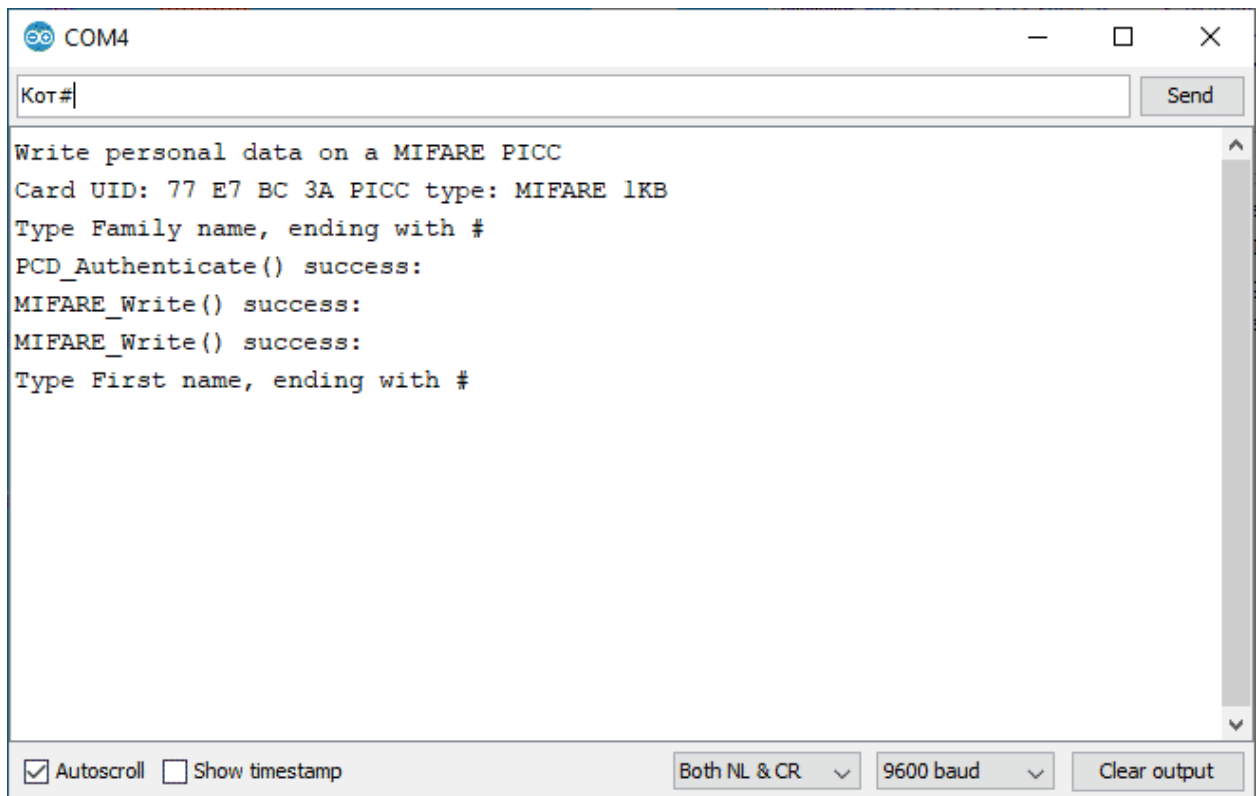


Рисунок 5.3 – Монітор послідовного порту

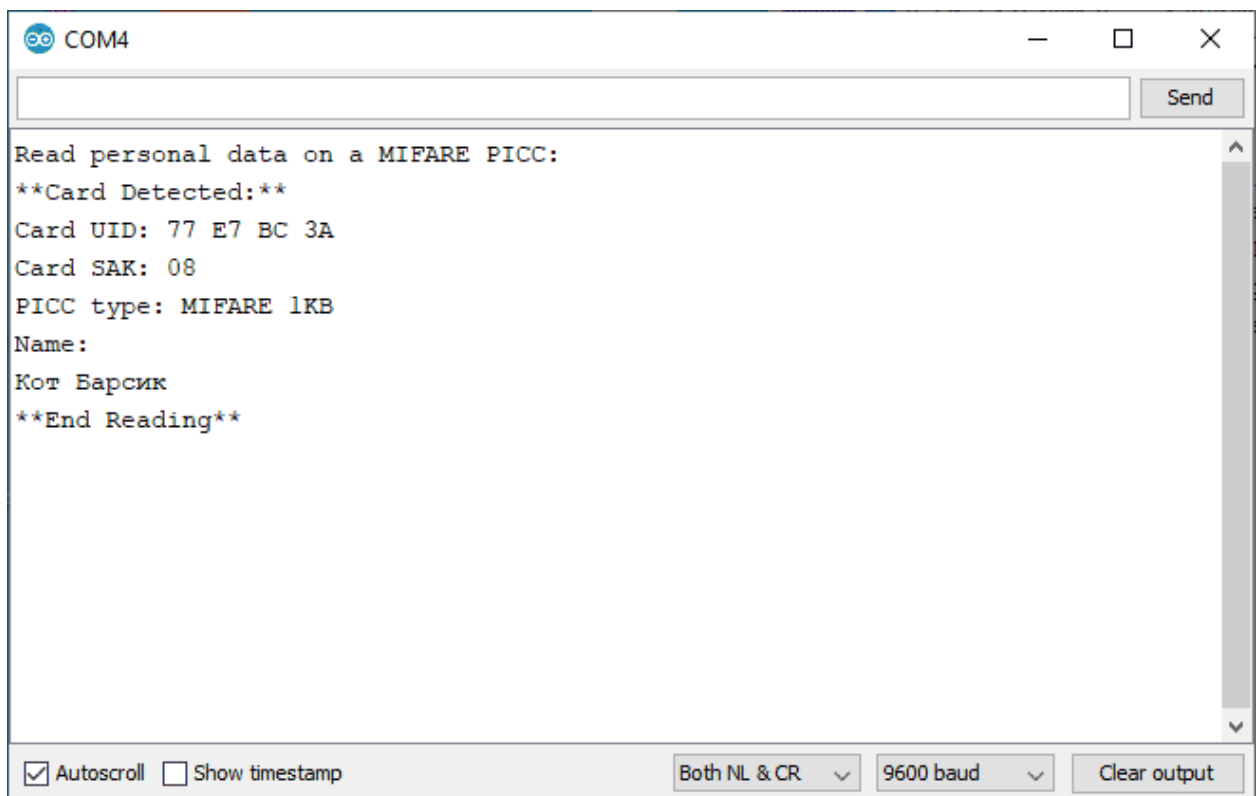


Рисунок 5.4 – Монітор послідовного порту після зчитування записаних даних

Функції бібліотеки:

- PCD\_DumpVersionToSerial () - виводить версію прошивки рідера в монітор порту;
- PICC\_IsNewCardPresent () - перевіряє наявність піднесеної мітки;
- PICC\_ReadCardSerial () - зчитує дані з мітки;
- PICC\_DumpToSerial () - виводить дані мітки в монітор порту;
- PICC\_GetType () - тип карти;
- PICC\_GetTypeName - ім'я карти.

\*Додаток 4 – приклади програми для роботи з модулем.

#### **5.4 Розробка Android – додатку в середовищі App Inventor 2**

MIT App Inventor - це інтуїтивне візуальне середовище програмування, яке дозволяє кожному - навіть дітям - створювати повнофункціональні програми для смартфонів та планшетів. Новачки в MIT App Inventor можуть створити просту першу програму, яка може стати працездатною менш ніж за 30 хвилин. Більше того, цей інструмент на основі блоків сприяє створенню складних програм із високим ступенем алгоритмізації за значно менший час, ніж традиційні середовища програмування. Проект MIT App Inventor прагне демократизувати розробку програмного забезпечення, надавши всім людям, особливо молоді, можливість перейти від споживання технологій до створення технологій [39].

Невелика команда співробітників та студентів CSAIL на чолі з професором Хелом Абельсоном формує ядро міжнародного руху винахідників. На додаток до провідних інформаційних програм навколо MIT App Inventor та проведення досліджень щодо їх впливу, ця основна команда підтримує безкоштовне середовище розробки онлайн-додатків, яке обслуговує понад 6 мільйонів зареєстрованих користувачів.

Почнемо з дуже простого додатка, яке буде відправляти через Bluetooth BLE два коди управління, "0" і "1". "0" буде служити командою для вимкнення світлодіода (LED), і "1" командою на включення LED. Це означає, що

програмі знадобиться якийсь спосіб відправити ці коди, і платі Arduino потрібно спосіб їх прийняти і обробити. Спочатку ми напишемо зовсім простий додаток, який нічого не знатиме про первісний стан LED. Додаток просто відправляє ці коди, коли користувач натискає на кнопки. Просунуті функції будуть додані пізніше.

Щоб спростити процес програмування, програма створювалася на основі шаблону BaseConnect, створеного командою AI2. Файл aia шаблону додатки BaseConnect потрібен для сканування і підключення пристроїв BLE. Є деякі незручності в такому рішенні (на зразок розміру тексту в списку пристроїв), проте це хороша точка для швидкого старту. При розробці додатків BLE не обов'язково використовувати файл BaseConnect, але його потрібно використовувати як зразок, якщо ми починаємо розробляти додаток BLE в середовищі AI2 [40].

Відкриваємо App Inventor, відкриємо aia-файл BaseConnect, і зберігаємо його як `ARD_HM10_AI2_Single_LED_01.aia`. Модуль HC-05 доступний для підключення на пристрої Android, якщо використовувати додаток MIT AI2 Companion. Натискаємо Scan, і побачимо, які пристрої можна знайти в мережі Bluetooth BLE.

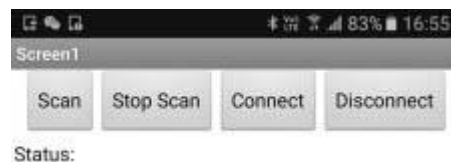


Рисунок 5.5 – Інтерфейс App Inventor BaseConnect

На рис. 5.6 видно різні пристрої BLE, включаючи HC-05 (перший у списку). На жаль, текст дуже малий і незручний для читання.

У Дизайнері (Designer) обираємо ListBLE, і змінюємо розмір тексту TextSize на 48. Тепер можна прочитати імена знайдених пристроїв BLE (див. рис. 5.7).

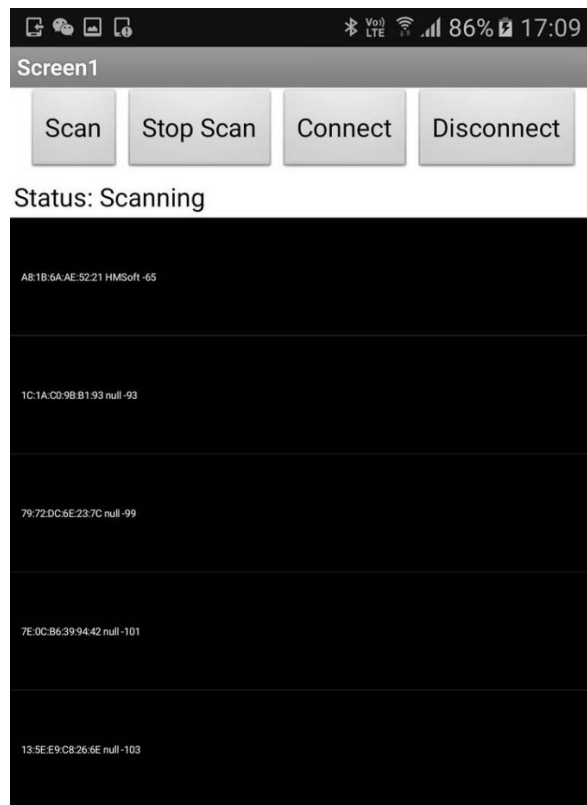


Рисунок 5.6 – Інтерфейс App Inventor BaseConnect після сканування BLE-пристроїв

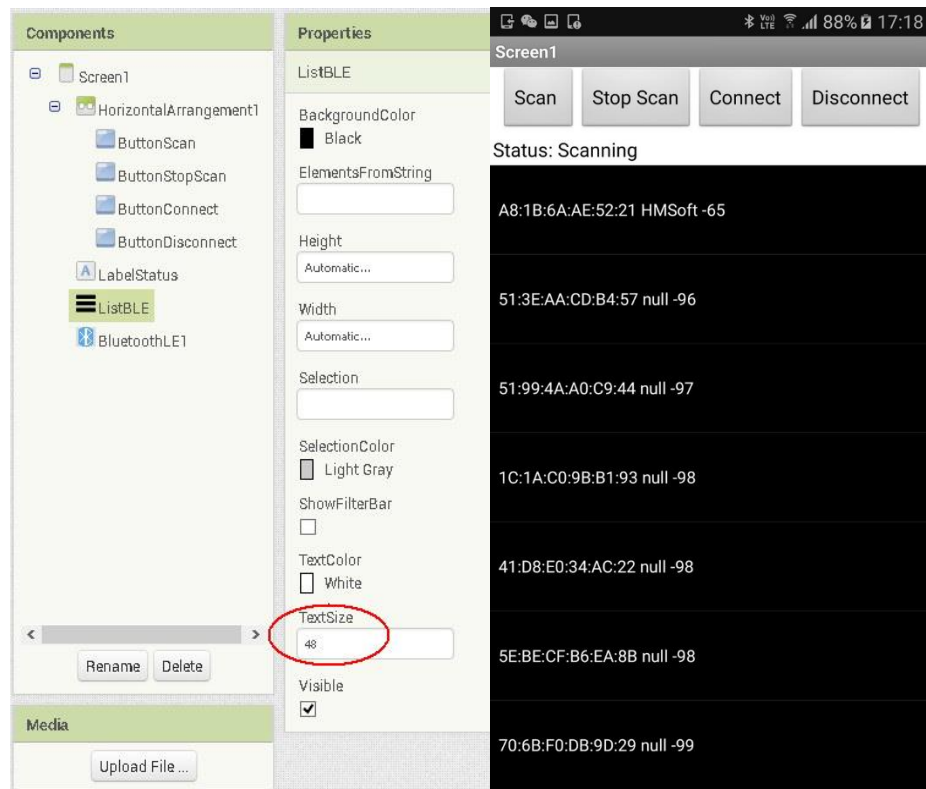


Рисунок 5.7 – Інтерфейс App Inventor BaseConnect після сканування BLE-пристроїв та зміни розміру шрифту

Додаток BaseConnect дуже простий, і в секції Blocks ми можемо побачити, що він складається із 7 блоків або процедур. Те, що робить кожна процедура, має бути самодокументованим і зрозумілим [41].



Рисунок 5.8 – Інтерфейс App Inventor BaseConnect. Секція Blocks

Додавання кнопок управління LED. Наступний крок - додавання органів управління світлодіодом, щоб можна було його включати і вимикати. Разом з кнопками, використовуємо текстові мітки з прогалін для організації інтервалів між кнопками. Кнопки розташовані горизонтально (Horizontal Arrangement), рис. 5.9. Потрібно перейменувати елементи, див. рис. 5.10.

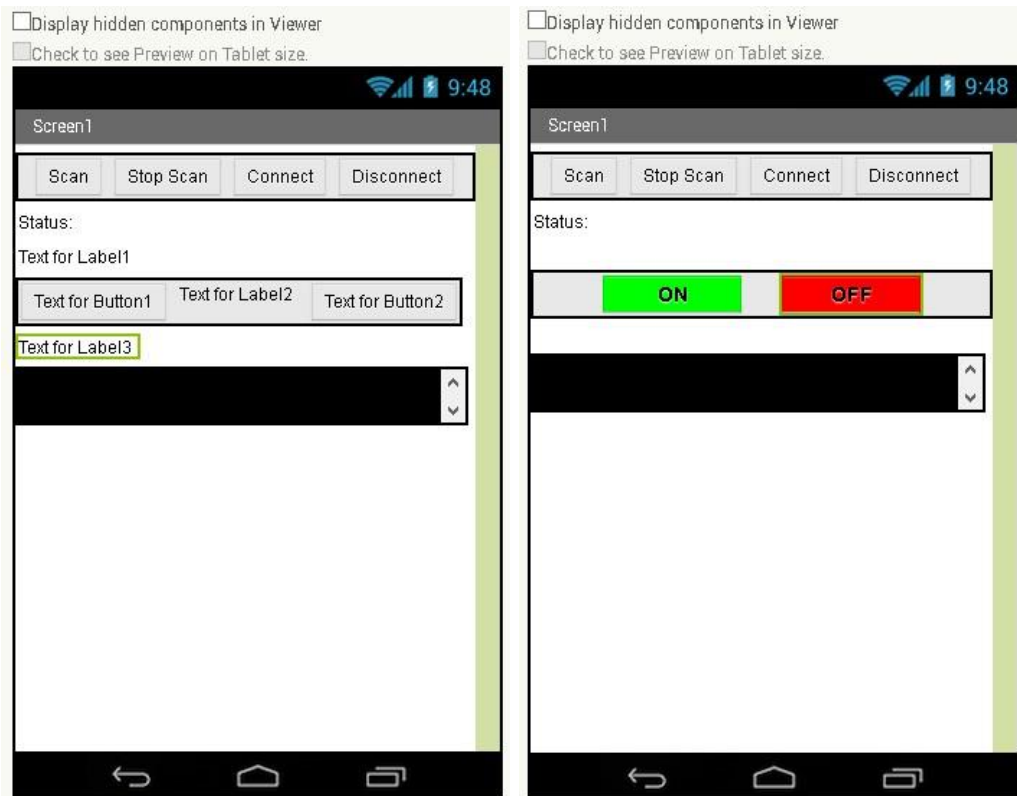


Рисунок 5.9 – Інтерфейс App Inventor BaseConnect після додавання кнопок керування LED

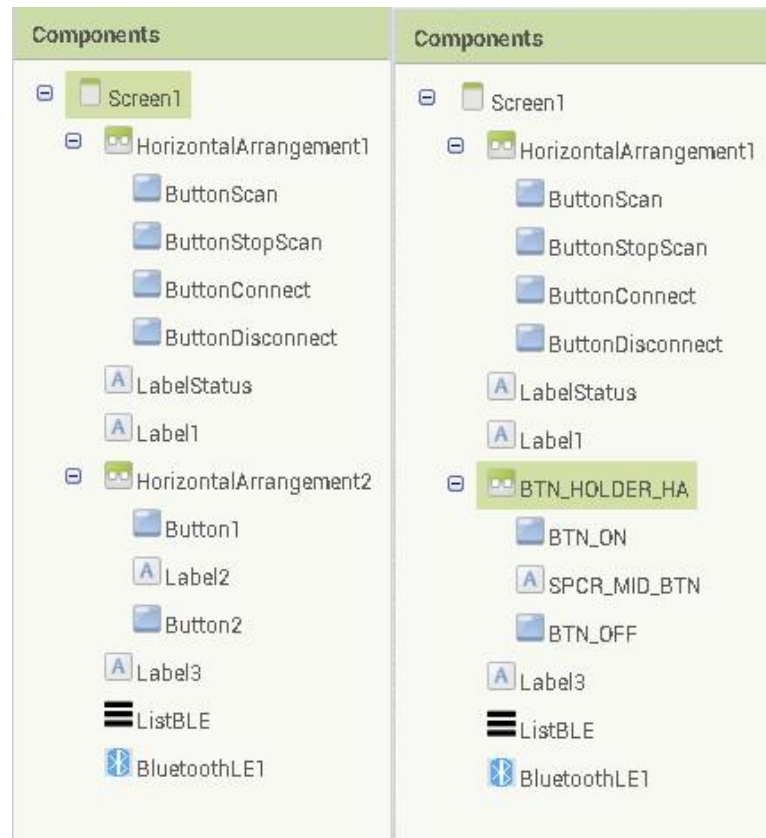


Рисунок 5.10 – Інтерфейс App Inventor BaseConnect після перейменування елементів

Після того, як ми додали елементи управління, зберігаємо проект в файл `ARD_HM10_AI2_Single_LED_02.aia`.

Тепер потрібно додати функції для відправки кодів управління. Це робиться не настільки прямолінійно, як можна було б очікувати. У протоколі Bluetooth Classic після того як було зроблено з'єднання, дані не просто записуються в канал з'єднання, у протоколі Bluetooth BLE потрібно записувати дані в спеціальну характеристику, і це означає, що нам потрібно знати коректний UUID для цієї характеристики, і також UUID служби, якій ця характеристика належить. У BLE немає каналу зв'язку, є характеристики, які можна читати і / або записувати (в залежності від властивостей характеристики).

Розширення AI2 BLE дозволяє нам записувати різні типи даних, і для цього прикладу ми могли б використовувати байти або рядки, тому що для передачі "0" і "1" підійде будь-який з цих варіантів. Використовуючи рядки, в майбутньому можна було б застосувати більш складні керуючі коди без внесення зайвих змін [42].

Блоки BLE Write зазвичай очікують дані, передані в списку. На щастя, якщо ми використовуємо прості дані (plain data), AI2 буде їх прозоро перетворювати в потрібний тип.

Щоб можна було записати дані, нам потрібно знати service UUID і characteristic UUID. Функція `WriteStrings` сліпа, вона просто відправляє дані, і у неї немає ніякого поняття про те, чи приймає їх HC-05, чи ні. У розширенні BLE ми можемо перевірити завершення передачі шляхом використання блоку `WriteStringsWithResponse`.

Він вимагає віжд модулю HC-05 відправити назад підтвердження. AI2 отримує це підтвердження, і це викликає спрацьовування події в AI2, яке ми можемо перевірити. Поки нам це не потрібно, будемо використовувати тільки функцію `WriteStrings`.



У редакторі блоків (Blocks editor), додаємо кілька глобальних змінних, щоб в них тримати номери UUID і додаємо блоки події кліка на кнопках управління LED, див. рис. 5.11.

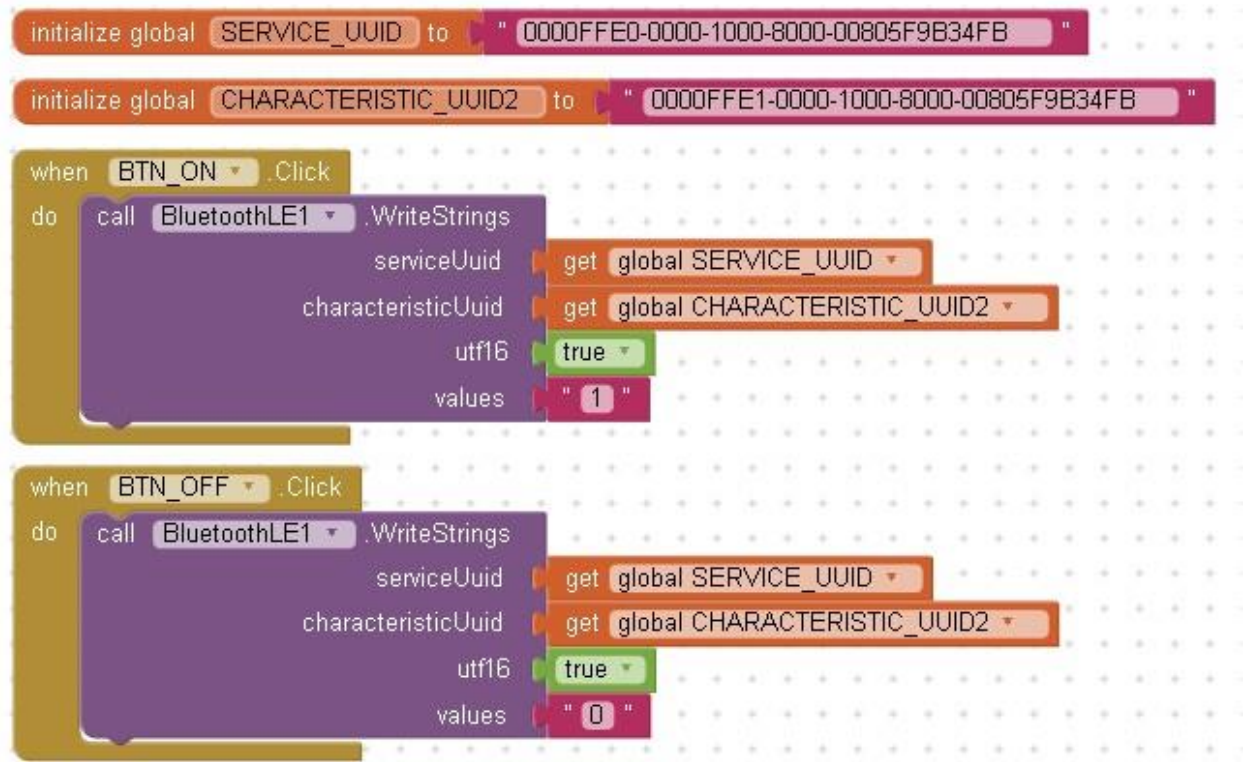


Рисунок 5.11 – Інтерфейс Blocks editor

Тепер нам потрібно реалізувати дії при натисканні кнопки ON - щоб надсилався код "1", і при натисканні на кнопку OFF надсилався код "0".

Додаємо Bluetooth Client і потім додаємо оповіщувач (notifier), див. рис. 5.12.

Тепер нам потрібно змінити процес, який буде відбуватися при натисканні на кнопку Scan. Коли на цій кнопці здійснено клік, нам спочатку треба перевірити, чи дозволена робота Bluetooth, якщо дозволена, то можна сканувати. Якщо немає дозволу - то нам потрібно видати повідомлення про помилку. Перевірка, чи дозволений Bluetooth, робиться за допомогою блоку `BluetoothClient.Enabled`.

У редакторі блоків додаємо перевірку Bluetooth в процедуру `ButtonScan.Click`. Одночасно додаємо блок `else` до блоку `if / then`, і викликаємо оповіщувач.

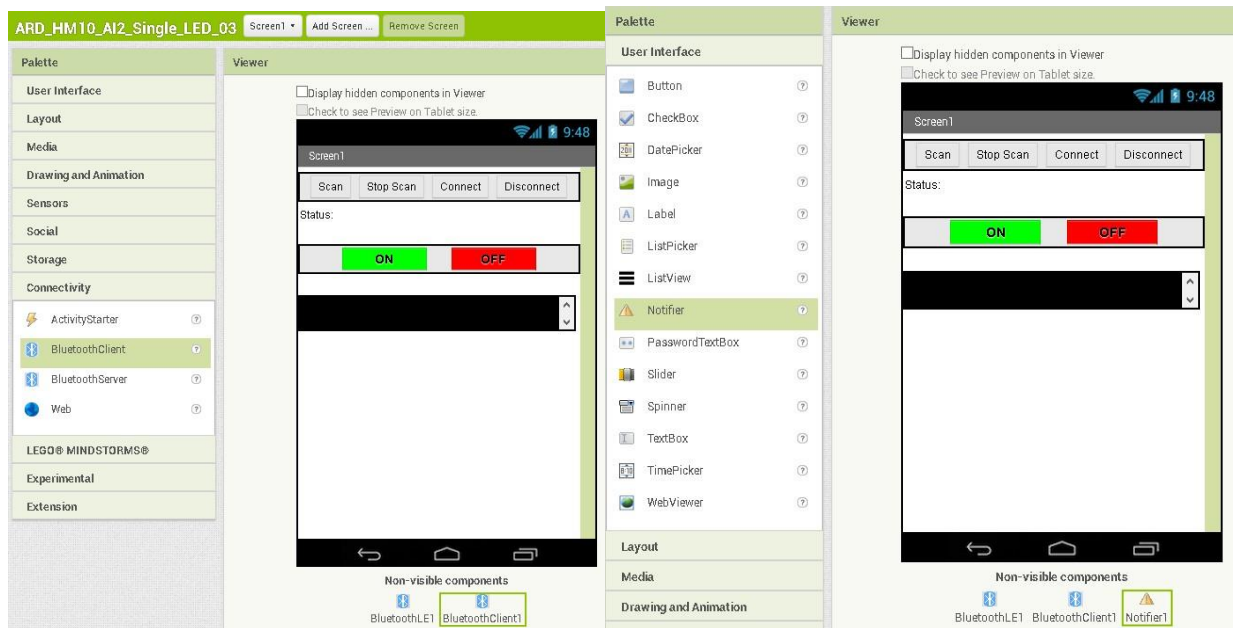


Рисунок 5.12 – Додавання Bluetooth Client

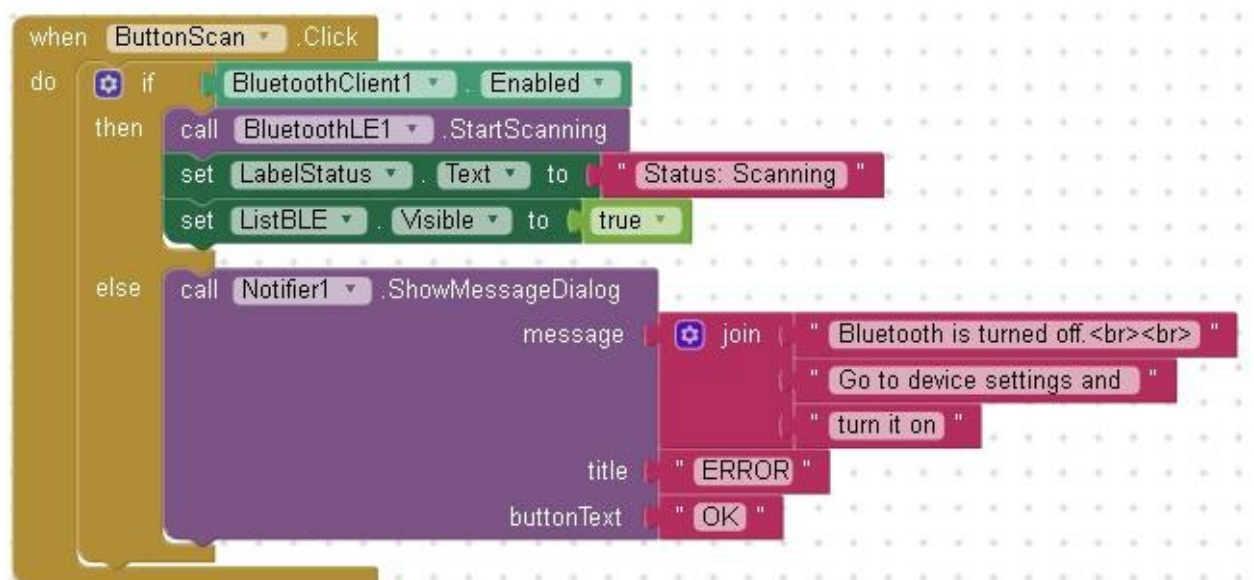


Рисунок 5.13 – Додавання BluetoothClient.Enabled та notifier

Тепер якщо ми спробуємо запуснути сканування перед включенням Bluetooth, то отримаємо попередження. З кнопками управління LED – нам потрібно перевірити, чи активне підключення, перед спробою відправки кодів управління "0" і "1". Це можна зробити в блоці BLE isDeviceConnected.

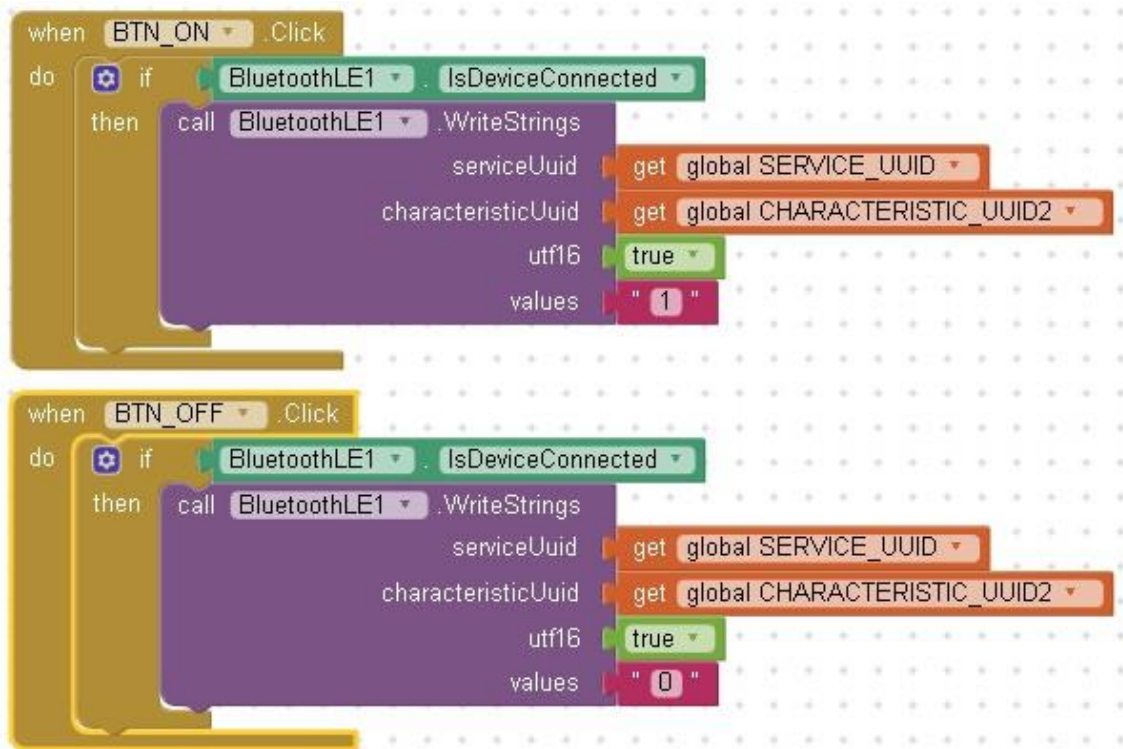


Рисунок 5.14 – Додавання BLE isDeviceConnected

Тепер якщо клікнути на кнопку ON або OFF, коли HC-05 не підключений, то нічого не станеться. Ми могли б, якби захотіли, відобразити повідомлення про помилку. Але швидше за все мати занадто багато подібних повідомлень не дуже зручно, тому краще нічого не робити. Як варіант, можна зробити ці кнопки неактивними до тих пір, поки не буде активовано підключення.

Відкриваємо програму ще раз, тепер вона буде виглядати більш якісно. Коли Bluetooth підключення не дозволено, ми отримаємо інше повідомлення, яке говорить нам, що потрібно його включити. Коли ми клікнемо на кнопки управління LED без наявності з'єднання, більше не будемо отримувати системне повідомлення про помилку. При кліці на інші кнопки з вимкненим Bluetooth ми отримаємо більше системних повідомлень про помилки.

Зараз у нас є одна кнопка для запуску сканування, і ще одна для зупинки сканування. Їх можна об'єднати однією кнопкою-перемикачем (toggle button), де буде змінюватися напис в залежності від поточного стану, SCAN і STOP SCAN. Додаток запуститься з написом на кнопці SCAN, і якщо на неї натиснути, то при включеному Bluetooth запуститься сканування, і на кнопці зміниться напис на STOP SCAN. Це означає, що користувач може натиснути

тільки на цю ж єдину кнопку з написом STOP SCAN, якщо він до цього натиснув на кнопку SCAN. Цей метод також означає, що текст на кнопці можна використовувати як якийсь прапор стану. Те ж саме можна зробити з кнопками Connect і Disconnect, замінивши їх на одну кнопку.

У Дизайнері змінюємо текст на кнопці Scan, і видаляємо кнопку Stop Scan. У редакторі блоків видаляємо функцію обробки події ButtonStopScan.Click, вона більше не потрібна.

Тепер при кліці функція спочатку перевіряє текст на кнопці, якщо цей текст "SCAN", то ми знаємо, що потрібно запустити сканування. Якщо текст не "SCAN", то цей текст повинен бути "STOP SCAN", і потрібно встановити сканування.

Те ж саме виконаємо з кнопками Connect та Disconnect. Об'єднаємо дві кнопки в один перемикач, який забезпечить можливість користувача виконати відключення тільки в тому випадку, якщо перед цим він активував підключення.

У елемента A12 ListView є кілька властивостей, які ми можемо використовувати, якщо обраний один з елементів списку: List.Selection і List.SelectedIndex. Перед тим, як зроблений вибір у списку, значення List.SelectedIndex дорівнює 0 і List.Selection одно null або "". Використовуючи SelectionIndex, ми можемо легко побачити, що якщо SelectionIndex більше 0, то користувач щось вибрав в списку.

На даний момент коли користувач клацає на кнопку CONNECT / DISCONNECT, на кнопці буде відповідним чином змінюватися текст. Він зміниться навіть в тому випадку, коли є проблема із з'єднанням, і воно з якоїсь причини не відбулося. Щоб виправити це, перемістимо інструкції зміни тексту в функції BluetoothLE1.Connected і BluetoothLE1.Disconnected. Це має виглядати як на рис. 5.16.



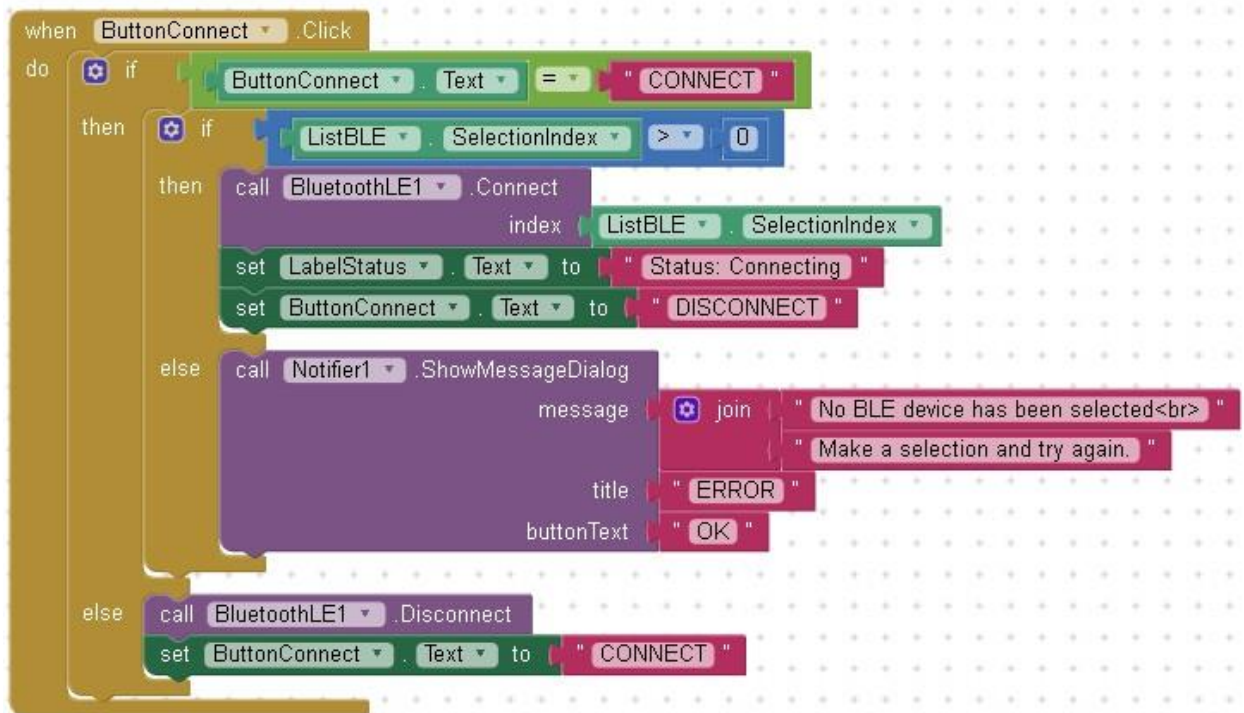


Рисунок 5.15 – Додавання List.SelectionIndex

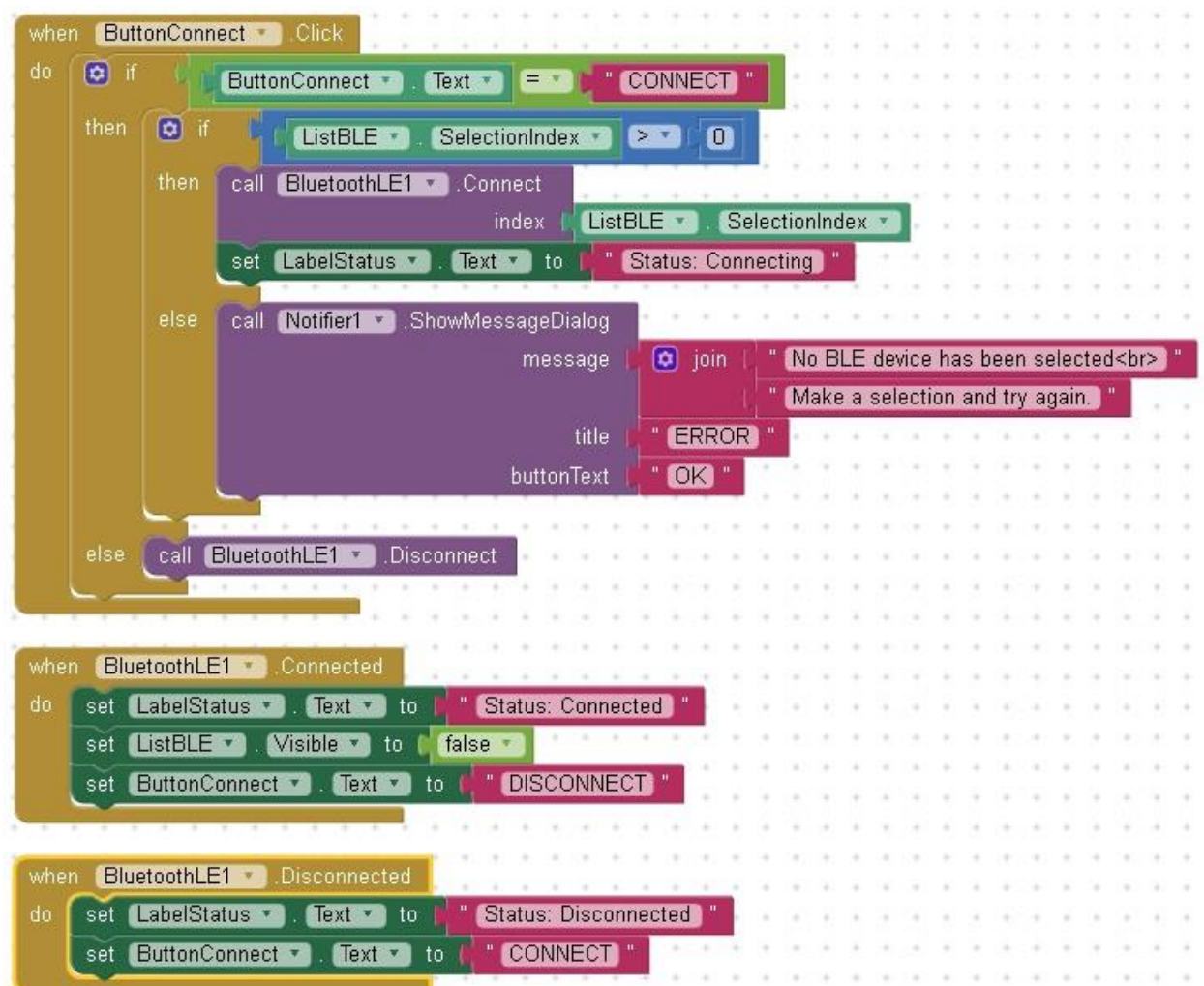


Рисунок 5.16 – Переміщення інструкцій зміни тексту

У функції обробки події `ButtonConnect.Click` зараз є інструкція для установки тексту `LabelStatus` в `"Status: Connecting"`, що можна також при бажанні зробити і з кнопкою `CONNECT / DISCONNECT`.

Зберігаємо поточний проект в `ARD_HM10_AI2_Single_LED_04.aia`.

Наступним кроком, звичайно буде комбінування двох кнопок управління LED в один перемикач, на якій буде текст або `"ON"`, або `"OFF"`.

Щоб об'єднати дві кнопки в одну ми виконаємо точно такі ж дії, які робили раніше. При кліці на кнопку потрібно перевірити текст, який на ній присутній. Якщо цей текст `"ON"`, то ми знаємо, що LED світиться, його потрібно вимкнути і поміняти текст на кнопці на `"OFF"`. В іншому випадку на кнопці буде текст `"OFF"`, LED вимкнений, нам треба його включити і поміняти текст на кнопці на `"ON"`.

Будемо використовувати кнопку включення, перейменуємо її в `BTN_LED`.

Слід зауважити, що кнопка управління LED зараз відображає стан LED, а не відправляє команду. Це означає, що відправлені коди треба змінити на зворотні. Раніше кнопка `ON` відправляла команду включення LED. Тепер все навпаки: якщо на кнопці напис `"ON"`, то це означає, що LED включений, і треба його вимкнути, тобто відправити команду виключення.

Можна тепер видалити кнопку `OFF` і порожні мітки, які використовувалися раніше для інтервалів між кнопками.

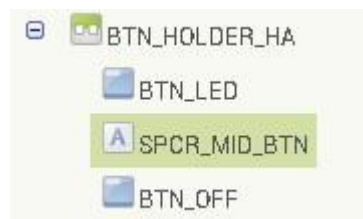


Рисунок 5.17 – Видалення зайвих міток

Після додавання оператора `if / then` для перевірки тексту на кнопці і переміщення блоків ми отримаємо код, зображений на рис. 5.18.

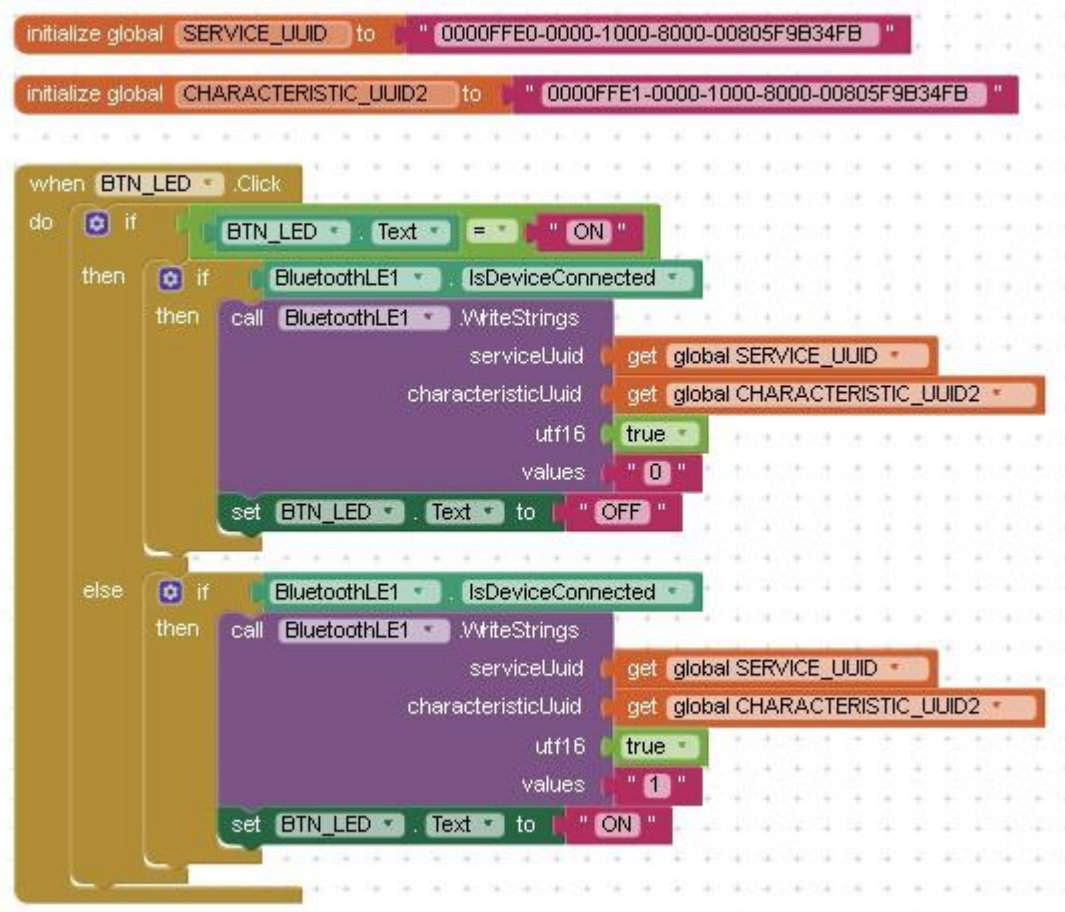


Рисунок 5.18 – Код після додавання оператора if / then для перевірки тексту на кнопці

Все відмінно працює, кнопка ON / OFF повністю керує світлодіодом LED. Однак є ще кілька речей, які добре б виправити:

- Коли додаток запускається, то на кнопці присутній текст "ON", хоча LED за замовчуванням виключений після подачі живлення на плату Arduino;

- Колір у кнопки не змінюється;

- Нам потрібно в двох місцях коду перевірити наявність з'єднання.

Змінюємо в Дизайнері текст кнопки на "OFF", і текст фону кнопки (background colour) на червоний (red). Тепер при старті зі стандартними програмами на кнопці буде текст OFF, і кнопка буде червона.

Щоб поміняти колір кнопки, ми будемо використовувати її властивість BackgroundColor. Кольори вибираються в меню Color.

Нижче показаний код з командами зміни кольору. Кнопка тепер разом з текстом буде міняти також і свій колір.

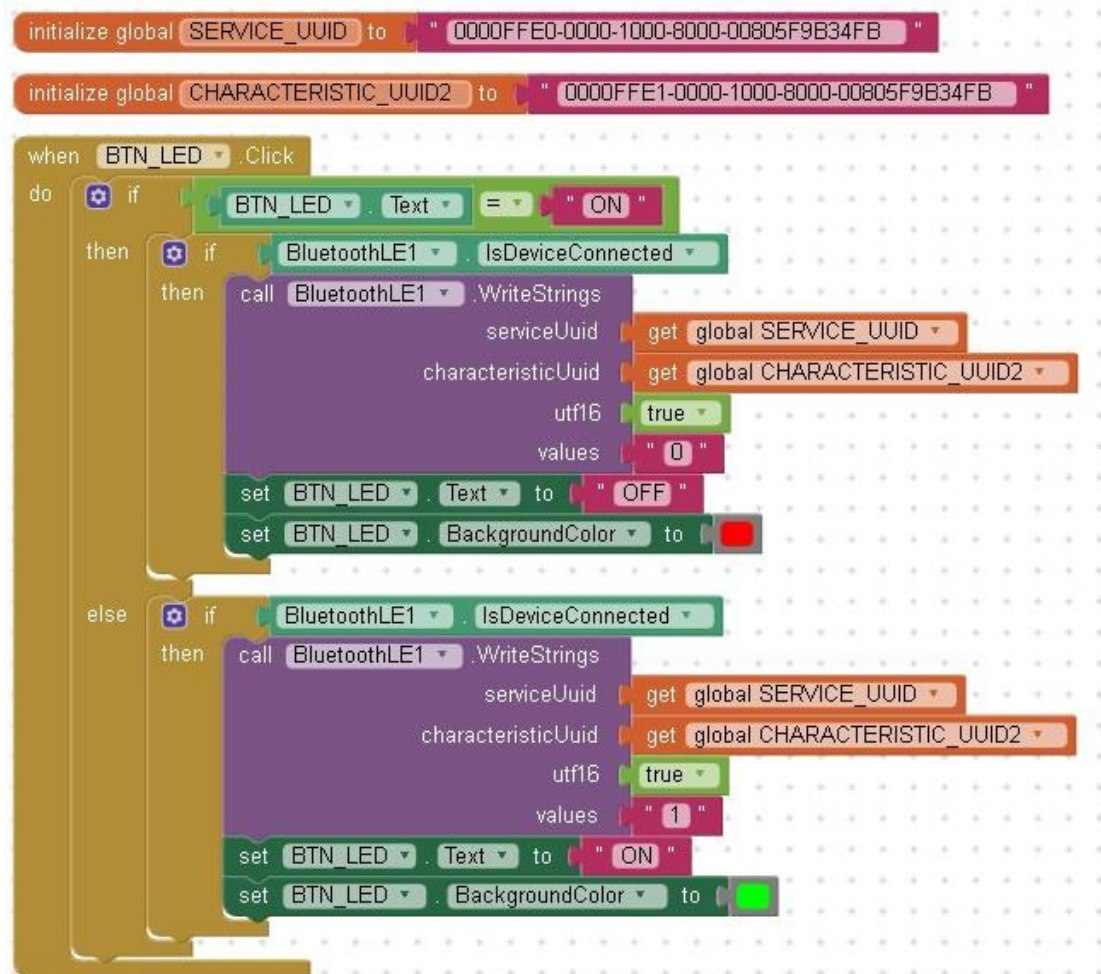


Рисунок 5.19 – Код після додавання функції зміни кольору

### Висновки до розділу

1. Для керування ДПС використовується досить проста та просунута бібліотека. З її допомогою можна забезпечити досконале, продумане керування двигуном, забезпечити стійкість до зони нечутливості ШІМ-сигналів, відкалібрувати двигуни.
2. Модуль Bluetooth HC05 водночас простий в налаштуванні, але містить ряд важливих параметрів, котрі обов'язкові до налаштування для коректного зв'язку пристрою з розробленим додатком із платою керування.
3. RFID модуль RC522 необхідний для організації ключового доступу до системи, та одного із зручних користувачу способу отримання/постановки авто на паркінг.



4. Розробка Андроїд-додатку – один із найбільш трудомістких процесів, котрий вимагає досить серйозних об’ємів знань та розуміння особливостей передачі даних за допомогою Bluetooth між пристроями. Зокрема, присутні особливості при налаштуванні зворотного зв’язку.

## РОЗДІЛ 6. ДОСЛІДЖЕННЯ РОБОТИ ЕЛЕКТРОМЕХАНІЧНОЇ СИСТЕМИ БАГАТОЯРУСНОЇ ПАРКІНГОВОЇ СИСТЕМИ АВТОМОБІЛІВ НА ЕКСПЕРИМЕНТАЛЬНОМУ СТЕНДІ

Поточний розділ покликаний дослідити сигнали, що передаються від контролера до виконавчих органів, у тому числі драйвери двигунів механізмів підйому та відвантаження. Окремо досліджується рівень сигналу від датчиків та вплив наявності електролітичних та кремнієвих конденсаторів для фільтрації сигналів, покращення завадозахищеності.

### 6.1 Рівні сигналів, що надходять від датчиків

#### *6.1.1 Індуктивний датчик перешкод*

Оскільки живлення датчика виконане від лінії 12 В, а принцип його роботи в спрощеному вигляді побудований на замиканні сигнального контакту із лінією «+» вхідного живлення, ми маємо знизити напругу до прийняттого рівня для контролера – максимум 5 В. Для цього скористаємося дільником напруги. [43].

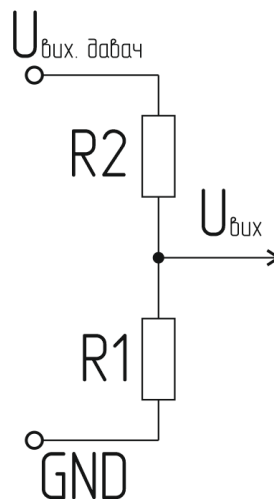


Рисунок 6.1 – Схема дільника напруги

Дільник напруги не є особливо надійним та точним способом зниження напруги, але використовується в системах із низьким споживанням, та з умовно широким допуском до рівня напруги. Застосовуємо дільник напруги, що базується на резисторах  $R1=1.2k$ ,  $R2=2k$ . У результаті дістаємо напругу на виході з датчика рівну 4.5 В.

### 6.1.2 Оптичний (інфрачервоний) давач перешкод

Робота інфрачервоного датчика наближення ґрунтується на випромінюванні інфрачервоного променя і прийманні його відбиття. Використовується широкосмуговий компаратор напруги LM393. Датчик має регульовану чутливість і оснащений індикатором вихідного сигналу.

У загальному – цей ІК-датчик наближення застосовується для прокладки шляху роботів, обходу перешкод, лічильника перешкод і визначення перешкод на шляху загалом. Головний недолік – чутливість до освітленості в приміщенні, що вимагає додаткового калібрування залежно від погодних умов та часу доби, або розміщення в умовно ізольованому боксі.

У нашому випадку – головна ціль давача, це контроль наявності автомобіля на платформі. Датчик має інверсований вихід, тому в разі спрацювання рівень сигналу на виході рівний нулю.

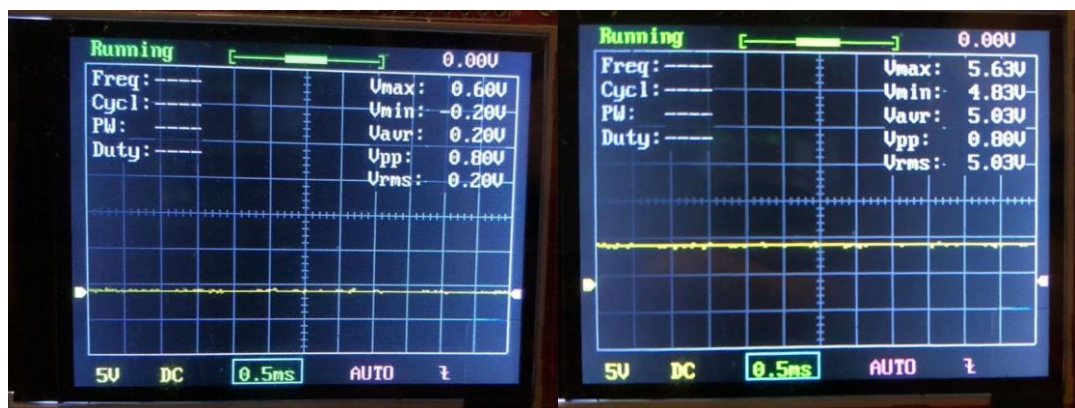


Рисунок 6.2 – Режим роботи оптодатчика

## 6.2 Дослідження роботи двигунів у випадку керування методом ШІМ у статичному режимі

ШІМ – є імпульсний сигнал постійної частоти та змінної шпаруватості, тобто відношення періоду проходження імпульсу до його тривалості. З допомогою завдання шпаруватості (тривалості імпульсів) можна міняти середню напругу на виході ШІМ. У цифровій техніці, виходи якої можуть приймати тільки одне з двох значень, наближення бажаного середнього рівня виходу з використанням ШІМ є абсолютно природним.

В роботі використано 2 драйвери для ДПС, 4 двигуни постійного струму, попарно однакового модельного ряду. Одна з пар двигунів оснащена редукторами для збільшення вихідного моменту на валу, зниження кількості обертів та не значних масогабаритних показників.

### 6.2.1 ШІМ-сигнал від Arduino до драйверів L298N

Перевіримо ШІМ від контролера до драйвера на рівнях сигналу від робочого мінімуму двигунів до максимального значення (орієнтовно – від 50 до 255 біт).



Рисунок 6.3 – ШІМ від контролера. Рівень завдання 50 та 100 біт



Рисунок 6.4 – ШІМ від контролера. Рівень завдання 150 та 200 біт

На рис. 6.3, графік ліворуч – можемо спостерігати шпаруватість приблизно 20 % на частоті орієнтовно 490 Гц. Діюче значення напруги знизилося до майже 2 В. На рис. 6.3, графік праворуч можемо спостерігати шпаруватість приблизно 40 % на частоті орієнтовно 490 Гц. Діюче значення напруги знизилося до майже 3 В.



На рис. 6.4, графік ліворуч – шпаруватість рівна порядку 60 % при сталій частоті орієнтовно 490 Гц та діючій напрузі приблизно 3,7 В, та на графіку ліворуч шпаруватість піднялася до майже 80 % за діючої напруги 4,3 В.

На рисунку 6.5 номінальне значення завдання та номінальне значення параметрів на виході з контролера.



Рисунок 6.5 – ШІМ від контролера. Рівень завдання 254 біт

*6.2.2 ШІМ-сигнал від драйверів L298N до двигунів механізму відвантаження автомобіля*



Рисунок 6.6 – ШІМ від драйвера. Рівень завдання 21 та 50 біт

На рис. 6.6 – 6.8 зображено осцилограми надходження імпульсів керування на двигун із редуктором механізму відвантаження. Очевидно, що робота двигуна спотворює форму імпульсів. Найбільш помітні відхилення після зниження рівня сигналу від драйвера до нуля, у цей момент діє самоіндукція у двигуні, котра вносить імпульсні завади в роботу драйвера.



Рисунок 6.7 – ШІМ від драйвера. Рівень завдання 100 та 150 біт



Рисунок 6.8 – ШІМ від драйвера. Рівень завдання 200 та 254 біт

*6.2.3 ШІМ сигнал від драйверів L298N до двигунів механізму відвантаження автомобіля після встановлення керамічного конденсатора ємністю 0.1 мкФ*

Після встановлення керамічного конденсатора насамперед помітно, що розширився діапазон регулювання двигуном внаслідок зменшення нижнього порогу «мертвої зони». Окрім того, робота двигуна стала більш стабільною (звук роботи став м'якший, оберти стійкіші), зменшилися пікові стрибки напруги.

Додатково було виміряно струми споживання двигуна за різних сигналів керування. Після досягнення номінальної швидкості – струм 30 мА, за 50 % швидкості (50 % шпаруватості / 125 біт сигнал) – 15 мА, за мінімально можливої швидкості – 10 мА.





Рисунок 6.9 – ШІМ від драйвера. Рівень завдання 13 та 100 біт



Рисунок 6.10 – ШІМ від драйвера. Рівень завдання 150 та 200 біт

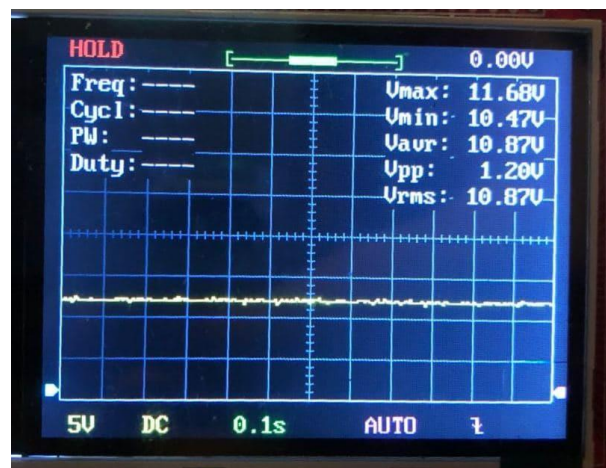


Рисунок 6.11 – ШІМ від драйвера. Рівень завдання 254 біт

#### 6.2.4 ШІМ-сигнал від драйверів L298N до двигунів підйомного механізму

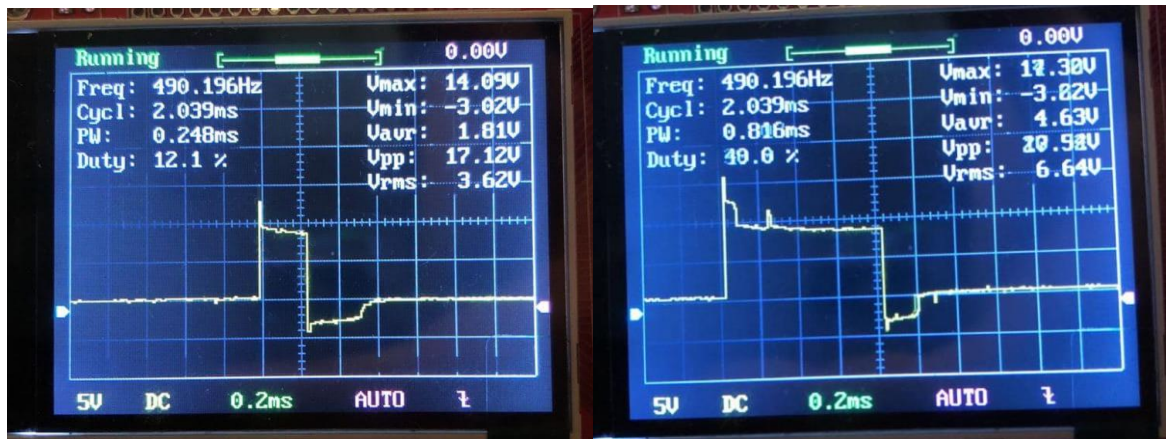


Рисунок 6.12 – ШІМ від драйвера. Рівень завдання 31 та 100 біт



Рисунок 6.13 – ШІМ від драйвера. Рівень завдання 150 та 200 біт

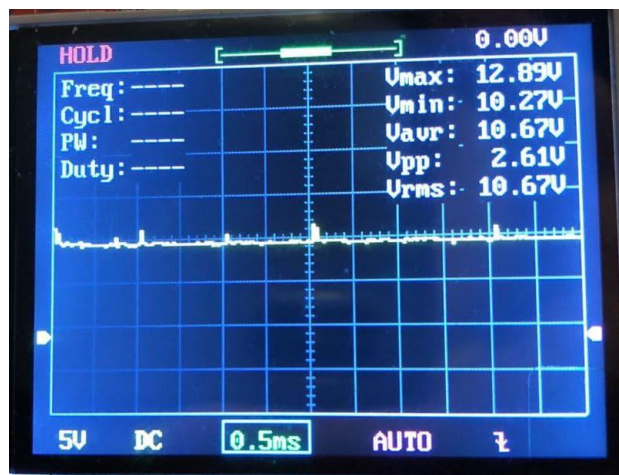


Рисунок 6.14 – ШІМ від драйвера. Рівень завдання 254 біт

На рис. 6.12 – 6.14 зображено осцилограми роботи двигуна постійного струму з редуктором підйомного механізму. Робота двигуна, очевидно, не стійка,



спостерігаються часті перепади рівнів сигналу, стрибки напруги. Спробуємо згладити їх за допомогою керамічного конденсатора.

*6.2.5 ШІМ сигнал від драйверів L298N до двигунів підіймального механізму автомобіля після встановлення керамічного конденсатора ємністю 0.1 мкФ*



Рисунок 6.15 – ШІМ від драйвера. Рівень завдання 35 та 100 біт



Рисунок 6.16 – ШІМ від драйвера. Рівень завдання 150 та 200 біт

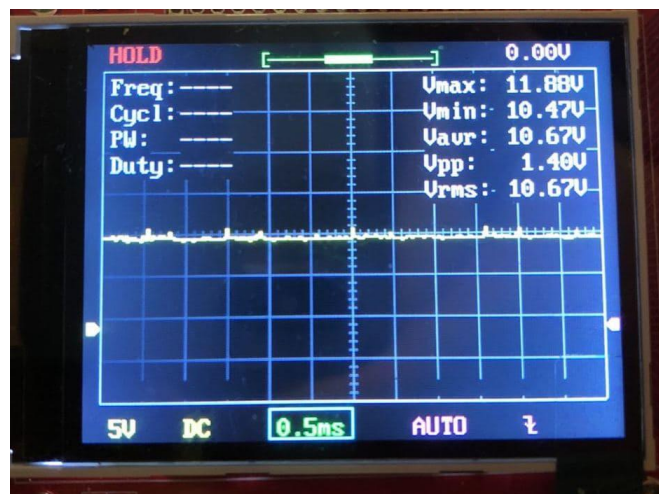


Рисунок 6.17 – ШІМ від драйвера. Рівень завдання 254 біт

Після встановлення керамічного конденсатора помітно згладилися сторонні пульсації, але загалом робота двигуна не є зразковою. Також зменшився мінімальний рівень пускового сигналу (підвищився рівень «мертвої зони»).

### **6.3 Регулювання ШІМ для досягнення плавного наближення до цільової позиції**

Головне завдання – виконувати пуск та гальмування ДПС найбільш плавно, але без втрати продуктивності. Важливість цих налаштувань насамперед набуває значущості в процесі наближення підйимального механізму разом з автомобілем до необхідного поверху, оскільки в разі різкого гальмування не просто можливий проїзд механізму повз точки початку відвантаження автомобіля, а і його пошкодження. Окрім того, у режимі імпульсного (різкого) приросту швидкості підвищується зношування як електричної, так і механічної частини системи. Першочергово розглядається два варіанти плавного регулювання:

— Двигун плавно набирає швидкість, досягає номінальної швидкості, у разі спрацювання датчика попереднього поверху (того, що на один рівень попереду від цільового) – швидкість знижується до певного комфортного значення.

У цьому випадку, механізм підйому проїде дистанцію від спрацювання датчика-лічильника попереднього поверху до спрацювання датчика-лічильника цільового поверху на зниженій швидкості, та зможе у разі спрацювання останнього зупинитися без ударів, розхитувань тощо.

— Траєкторія руху розраховується за декількома базовими алгоритмами, та враховує час, необхідний для проходження механізмом підйому певної дистанції (1 поверх, два, три).

У цьому випадку система орієнтується лише щодо вихідної та цільової позицій двигуна. Наприклад, завдання – піднятися на два поверхи, для цього треба витратити орієнтовно 10 секунд. Дві секунди – плавний розгін (підвищення швидкості), 6 секунд – рух на номінальній швидкості, 2 секунди на гальмування (зниження швидкості). У цьому випадку датчик виконує роль

кінцевого вимикача, і у разі його спрацювання двигун уже має рухатися на заздалегідь зниженій швидкості.

#### **6.4 Вплив зміни максимальної швидкості руху двигунів на точність позиціонування, аварійність тощо**

Завдання паркінгової системи – забезпечити не лише безпеку зберігання та транспортування автомобіля, а й належна швидкість завантаження та повернення авто власнику. Важливо знайти межу між швидкістю та безпечністю. Збільшення швидкості тягне за собою збільшення ривків за не змінних параметрах траєкторії. Посилені ривки, своєю чергою, зменшують як термін експлуатації паркінгової системи, так і вносять ризик пошкодити автівку клієнта.

Змінивши траєкторію для зменшення ривків – ми підвищуємо загальну швидкість роботи системи, але маємо слідкувати за тим, щоб оновлена траєкторія встигала відпрацьовуватися на коротких ділянках маршруту переміщення механізмів. Так, ми можемо збільшити номінальну швидкість у випадку пересування підйимального механізму між декількома поверхами, слідкуючи за тим, на скільки справляється із новими параметрами механічна частина. У такому випадку збільшення швидкості є доцільним.

Натомість – збільшивши номінальну швидкість на коротких ділянках – двигуни ризикують просто не встигнути розігнатися до номінальної швидкості, що спричинює аварійний режим і перехід до гальмівного режиму роботи.

Щодо роботи механічної частини у разі підвищення номінальної швидкості – є ризики того, що канат, який підіймає механізм, почне заплутуватися, унаслідок чого виникне перекошування положення платформи, що, своєю чергою, тягне за собою не точність у позиціюванні, заклинювання напрямних, не коректне відвантаження автомобіля з/на паркомісце, та, у найгіршому випадку, скидання авто з палети.

## **6.5 Розгляд можливості позиціювання механізмів без використання давачів (аварійний режим)**

Як було описано в п. 6.3, ми можемо використовувати часові розрахунки для відпрацювання траєкторії, але повноцінне позиціювання на основі часових інтервалів – виключно аварійний режим роботи системи. У цьому випадку, потрібно внести дельту часу, за який система має завершити певний етап процедури паркування автомобіля. Якщо цього не відбулося – у системі сталося відхилення від нормальної роботи, а, значить, необхідна подальша діагностика та виправлення несправності.

Якщо ж таки розглянути варіант бездавачевого позиціювання – необхідно провести ряд дослідів з замірами необхідного часу на проходження певних етапів маршруту, обрати середнє значення, підібрати адаптовані під цей режим роботи значення траєкторії (механізм певно має проходити маршрут зі стабільно прогнозованою швидкістю).

У разі невірогідності виконаних розрахунків та вибору параметрів можливі такі наслідки:

— Не значний переліт механізму підйому щодо цільової точки позиціювання у разі завантаження автомобіля.

Якщо переліт стався у разі підйому авто на цільовий поверх – небезпека умовно не висока, оскільки механізм відвантаження безпечно доставить автомобіль на повний виліт, після чого механізм підйому плавно спуститься вниз, що дасть можливість палеті з авто залишитися на спеціальних тримачах, після чого механізм відвантаження повернеться у вихідну позицію. У разі значного перельоту – вірогідно, що в разі спускання механізму підйому палета з автомобілем не зможе запаркуватися на відведених тримачах, а механізм відвантаження поверне авто назад на механізм підйому.

Переліт палети більш небезпечний у разі розгляду ситуації з накопичення похибки, які без давачів не може бути компенсованою. У такому випадку можливе розміщення контрольного «нульового» давача, який може

використовуватися для обнулення похибки після кожного циклу завантаження/розвантаження авто.

— Не значний переліт механізму підйому щодо цільової точки позиціювання при відвантаженні (поверненні) автомобіля.

У цій ситуації можливе вдаряння механізму відвантаження автомобіля об палету чи сам автомобіль, чи часткове зтягування палети з місця розміщення її на спеціальних тримачах. Будь-який варіант є небезпечним.

### **Висновки до розділу**

1. Два типи датчиків, що використовуються для контролю стану системи – інверсні, тобто нормально мають високий рівень сигналу на виході. Для підключення вихідного сигналу індуктивних датчиків до контролера необхідно знизити напругу з 12 В до прийнятних 4,5-5 В. Задля цього буде використано проста схема дільника напруги.

2. Контролер надсилає ШІМ сигнали на частоті близької до 500 Гц. Встановлення керамічних конденсаторів є необхідною мірою для покращення режимів роботи двигунів механізму підйому та відвантаження.

3. Є можливість підвищити максимальну швидкість роботи двигунів підйомального механізму лише на тривалих переїздах. В іншому випадку – є ризики зменшення як надійності системи. Так і втрата безпечної плавності та точності в роботі.

## РОЗДІЛ 7. СТАРТАП ПРОЄКТ

### 7.1 Мета проєкту

Залучення інвестиційних і кредитних коштів для будівництва та подальшої експлуатації автостоянки (багаторівневого автоматизованого паркінгу).

### 7.2 Актуальність проєкту

За браком достатньої кількості паркінгів - проблема, яка переслідує Київ вже не один рік. Особливо складна ситуація спостерігається в центральній частині міста: автомобілісти змушені паркуватися або уздовж тротуарів, тим самим створюючи масу незручностей для пішоходів, або уздовж дороги, що ускладнює рух, особливо на вузьких ділянках доріг.

За даними комунального підприємства «Київтранспарксервіс», у великих містах Європи кількість паркувальних місць в середньому становить приблизно 16-17% від кількості жителів міста. З огляду на європейський досвід, потреба Києва в паркувальних місцях становить не менше 500 000 машино-місць, в той час, як забезпеченість міста машино-місцями є удвічі нижче цього показника [44].

Рішення проблеми нестачі майданчиків для розміщення автотранспорту в місті через будівництво різних типів паркінгів передбачено Генеральним планом розвитку Києва до 2025 року та Програмою розвитку єдиного паркувального простору в Києві до 2015 року. Зокрема, в Генплані Києва одним з основних напрямів розвитку транспортного комплексу міста значиться будівництво підземних і наземних багаторівневих паркінгів у житлових масивах і біля транспортних пересадочних вузлів. Вирішувати питання паркінгів муніципалітет має намір не тільки за допомогою будівництва нових паркінгів, а й через зменшення кількості автомобілів в місті.

Програма розвитку єдиного паркувального простору в Києві до 2015 року, яка була затверджена 22 травня 2013 року Київською міською радою, передбачає будівництво багаторівневих і перехоплювальних паркінгів,

організацію паркувальних місць на прибудинкових територіях, впорядкування роботи автостоянок та гаражних кооперативів [45].

Таблиця 7.1. Паркувальний індекс деяких торгово-розважальних центрів Києва

Назва ТРЦ	GLA, кв.м	кількість машино-місць	паркувальний індекс
SKY MALL	67 000	2 500	3,7
«Караван»	45 300	1 300	2,9
Ocean Plaza	72 200	1 800	2,5
Dream Town	45 000	860	1,9

Джерело: Colliers International (Україна)

Таблиця 7.2. Деякі нормативи забезпеченості місцями для стоянки автомобілів в різних сегментах нерухомості, згідно з ДБН 360-92 \*\*

Деякі будівлі і споруди масового відвідування	Одиниця виміру	кількість машино-місць
Міністерства, установи управління, громадські, наукові та проєктні організації з великим числом відвідувачів	У розрахунку на 1000 працюючих	15-20
Міністерства, установи управління, громадські, наукові та проєктні організації з незначним числом відвідувачів	У розрахунку на 1000 працюючих	7-10
Вищі та середні навчальні заклади	У розрахунку на 100 осіб (викладачі, персонал і учні)	
Підприємства торгівлі та громадського харчування:		
Ресторани і кафе міського значення	У розрахунку на 100 місць в залах	10-15
Інші ресторани і кафе	У розрахунку на 100 місць в залах	8-12
Торгові центри, універмаги, магазини площею торгових залів понад 500 кв.м	У розрахунку на 100 кв.м торгової площі	8-12
криті ринки	У розрахунку на 50 торгових місць	20-25
Заклади культури та мистецтва:		
Театри, цирки, кінотеатри в центральній частині міста, концертні зали, музеї, виставки (міського значення)	У розрахунку на 100 місць або одночасних відвідувачів	10-15
Інші кінотеатри	У розрахунку на 100 місць або одночасних відвідувачів	5-7

Продовження табл. 7.2. Деякі нормативи забезпеченості місцями для стоянки автомобілів в різних сегментах нерухомості, згідно з ДБН 360-92 \*\*

Деякі будівлі і споруди масового відвідування	Одиниця виміру	кількість машино-місць
Готелі:		
високі категорії	У розрахунку на 100 місць розміщення	10-15
Інші готелі	У розрахунку на 100 місць розміщення	6-8
Лікувально-профілактичні заклади:		
Лікарні, диспансери, пологові будинки	У розрахунку на 100 ліжок	10-15
поліклініки	У розрахунку на 500 відвідувань в зміну	10-15
Спортивні будівлі і споруди міського значення з трибунами місткістю: стадіони - понад 5000 глядачів, зали і басейни - понад 500 глядачів	У розрахунку на 100 місць	3-5
Вокзали залізничного, річкового, морського, автомобільного і повітряного транспорту	У розрахунку на 100 пасажирів, які прибувають в годину пік	8-15

Примітка 1. Мінімальні норми в п.п. 3, 5, 6 таблиці дані для міст з порівняно низьким рівнем автомобілізації на розрахунковий термін (100-150 автомобілів на 1000 жителів), максимальні - для міст з порівняно високим рівнем автомобілізації (180-250 автомобілів на 1000 жителів).

Примітка 2. На відкритих стоянках автомобілів біля установ культурно-побутового обслуговування, підприємств торгівлі та відпочинку, певних будівель і споруд масового відвідування треба виділяти місця для особистих автотранспортних засобів інвалідів, позначаючи їх спеціальною розміткою і спеціальними знаками. Їхня місткість визначається залежно від загальної місткості автостоянки і складає: до 100 автомобілів - 4 місця-стоянки для інвалідів, від 100 до 200 - 5-7 місць стоянок, понад 200 - за розрахунком. Для лікувально-профілактичних установ, відвідуваних інвалідами у разі їхнього амбулаторного лікування, кількість машино-місць розраховується як 10-15% від загальної місткості автостоянки.



Проектування нових і реконструкція наявних гаражів і стоянок автомобілів (паркінгів) на території України регламентується нормативним документом ДБН В.2.3-15: 2007 «Автостоянки і гаражі для легкових автомобілів». Даними нормами визначаються вимоги до об'ємно-планувальних рішень автостоянок і гаражів, а також до їхнього інженерного забезпечення [46].

***Реалізація даного проєкту дасть можливість:***

- Вийти на ринок з вже відомим різновидом послуг: послуги надання машино-місць в оренду;
- Створити нові робочі місця в районі реалізації проєкту;
- Задовольнити попит на цивілізовані місця для паркування і місця для постійного зберігання автомобілів в районі реалізації проєкту;
- Поліпшити дорожньо-транспортну ситуацію завдяки забезпечення можливості паркування автомобілів в робочий час, а також постійного зберігання автомобілів мешканців прилеглих кварталів;
- Максимальне використання площі відведеного для будівництва ділянки;
- Забезпечити зручність заїздів і виїздів паркінгу;
- Нескладна, безпечна і зручна організація руху всередині (автоматизовано) і зовні паркінгу;
- Мінімальні витрати на утримання будівлі;
- Мінімальний питомий показник, що характеризується співвідношенням загальної площі гаража (стоянки) до його місткості.

### **7.3 Характеристика концепції проєкту**

Місце для паркування (багаторівневий паркінг) може бути розташоване в будь-якому регіоні України в місті з високим рівнем автомобілізації. Пропонований проєкт розроблений на прикладі міста Київ. Основним чинником, що впливає на рентабельність паркінгу, є географічне розташування. Найбільш економічно ефективне місце розташування залежить від призначення створюваного стоянкового комплексу. У разі створення стоянки для

зберігання автомобілів жителями міста, паркінг має розташовуватися в житловому мікрорайоні в безпосередній близькості від потенційних споживачів. У разі створення перехоплювальної стоянки автомобілів, її місце розташування визначається ключовими вузлами пересадки жителів з особистих автомобілів на громадський транспорт (метро) у зв'язку з високим завантаженням доріг.

У столиці є 3 зони паркування: перша – центр міста, друга – навколоцентральної частина та третя – спальні райони. Паркувальні талони - один з трьох законних способів оплати паркування в Києві. Вони бувають трьох типів. I - дає право на паркування в будь-яких зонах, II - на паркування в другій і третій зоні і III - відповідно в третій. Вартість талонів: квартальний на всі зони - тисяча чотириста шістьдесят п'ять гривень, місячний для всіх зон - 630 гривень, тільки для II і III зони - 441 гривня і тільки для III - 315 гривень.

Передбачається будівництво перехоплювальної стоянки, як одного з найбільш затребуваних типів стоянок.

*В результаті, розглядається три варіанти розміщення споруди автоматизованого паркінгу:*

— Центральна частина міста. Для неї характерні: висока орендна плата або вартість ділянки у разі покупки у власність, висока прохідність.

— Спальні райони міста. Вартість ділянки або орендна плата буде нижче, ніж в першому випадку. Потік клієнтів великий. Допускається розглянути ділянку поруч з новобудовами.

— На околиці міста. Місце характеризується низькою платою за оренду. Недолік - маленький людино-потік.

*За основу проєкту беремо такі вхідні дані:*

- Формат створюваного споруди: автостоянка / паркінг;
- Формат паркінгу за кількістю ярусів: багаторясний;
- Формат паркінгу за кількістю місць зберігання: середньої місткості;

- Формат паркінгу за кількістю рядів зберігання автомобілів: однорядна схема розташування автомобілів (забезпечує в'їзд і виїзд будь-якого автомобіля в будь-який час);
- Формат паркінгу за розташуванням місць зберігання автомобілів щодо проїздів: двосторонній (розташування місць зберігання уздовж обох протилежних сторін проїзду);
- Формат паркінгу за кутом установки автомобіля до поздовжньої осі проїзду: паралельний;
- Можливість використання двох незалежних підйомників з різних сторін паркінгу;
- Можливість довільного планування в'їзного простору в паркінг;
- Сегмент створюваного паркінгу: демократичний сегмент ринку, орієнтований на клієнтів із середнім рівнем доходу (передбачається забезпечити рівень цін на оренду машино-місця в 1,5-2 рази нижчі за середні, тому що саме в цьому сегменті спостерігається гостра нестача доступних паркувальних місць);
- Спеціалізація і послуги паркінгу: перехоплювальна стоянка, постійне зберігання автомобілів жителями довколишніх кварталів;
- Графік роботи паркінгу: цілодобово, щодня;
- Місткість паркінгу: до 100 машино-місць;
- Висота паркінгу: до 26500 мм;
- Загальна площа паркінгу: до 2500 м<sup>2</sup>;
- Персонал паркінгу: загальний штат персоналу - 6 осіб, із загальним фондом заробітної плати 100 тис. грн. у місяць;
- Вартість послуг паркінгу:
  - \* Вартість оренди паркінгу на постійній основі - 521 грн. в місяць за машино-місце;
  - \* Тривале паркування автомобілів – 170 грн. за машино-місце за три дні;

- \* Паркування автомобілів за сутки – 100 грн.;
- \* Погодинне паркування автомобілів - 25 грн. за машино-місце на годину;

Клієнтам, які залишають автомобіль на довгий термін, робиться знижка. Автовласникам пропонуються абонементи терміном від 1 до 6 місяців.

Для отримання додаткового прибутку планується надавати послуги із:

- зберігання шин;
- захист автомобіля спеціальним тентом;
- підкачування коліс;
- миття лобового скла;
- розморожування замків;
- прогрів машини до приходу клієнта;
- наявність точки швидкого харчування (фаст-фуд, напої).

Наявність супутніх послуг підвищить конкурентоспроможність і рентабельність стоянки. Після виходу в нуль і досягнення окупності проєкту можна відкрити міні-мийку поруч зі стоянкою [47].

## 7.4 Фінансування проєкту

### *Інвестиції*

— Загальний обсяг інвестицій в проєкт складає 3 727 750 грн. з урахуванням регулярних витрат за перший місяць. У структурі інвестиційних витрат найбільша частка припадає на обладнання і оснащення паркінгу - 94.61% (див. табл. 7.3 – 7.4).

- Тривалість інвестиційного періоду : 5 місяців.
- Початок роботи паркінгу: 6 місяць реалізації проєкту.
- Вихід на проєктну потужність: 11 місяць реалізації проєкту.

Передбачається фінансування проєкту на 20% використовуючи власні кошти ініціатора проєкту або власні кошти інвестора і на 80% завдяки позиковим грошовим коштам.

### *Система оподаткування*

У проєкті передбачається використовувати стандартну систему оподаткування.

Ця система оподаткування передбачає виплату таких податків:

- Податок на майно – 62,59 грн/м<sup>2</sup>;
- Податок на додану вартість – 20%;
- Єдиний соціальний внесок - ставка залежить від категорії працівників, основна ставка - 22% від нарахованої зарплати;
- Податок на прибуток фізичних осіб (ПДФО) - податок на нараховану ЗП і подібні платежі. Ставка - 18%;
- Військовий збір (ВЗ) - тимчасовий збір на доходи, що підлягають оподаткуванню ПДФО. Ставка ВЗ - 1,5%.

Згідно з бізнес-планом стоянки, що потрібно, щоб облаштувати територію автостоянки:

- Вирівняти ділянку, засипати її гравієм або укласти асфальт;
- Передбачити зручні під'їзні шляхи до автостоянки;
- Побудувати огорожу або сітку за периметром;
- Організувати електропостачання;
- Встановити ворота або шлагбаум;
- Організувати відеоспостереження за ділянкою;
- Облаштувати злив;
- Провести водопостачання;
- Поставити на паркінгу робоче місце для охоронців з одним кабінетом (студією);
- Організувати робоче місце керівного персоналу, та інженерів.

Таблиця 7.3 – Щомісячні витрати на утримування автоматизованого паркінгу

Витрати	Сума (грн.)
Оренда території	54 000
Упорядкування території	6 000
Обладнання, екіпіювання та утримання персоналу	5 000
Реклама	30 000
Зарплата персоналу	88 000
Інші витрати	20 000
Сумарно:	177 000

Таблиця 7.4 – Обладнання, його вартість

Найменування	Кількість	Ціна за од., грн	Сума, грн
Відеоспостереження	1	25 000	25 000
Роб. місце охоронця	1	40 000	40 000
Шлагбаум	1	25 000	25 000
Стілець	7	750	5 250
Шафа	3	2 000	6 000
Мікрохвильовка	1	3 500	3 500
Чайник	1	2 000	2000
Будівництво паркінгу (враховуючи обладнання автоматизації)	1	3 420 000	3 420 000
Сумарно:			3 526 750

## 7.5 Співробітники і персонал

Приклад бізнес-плану автостоянки містить штатний розпис співробітників, яких потрібно найняти:

- охоронці (3 співробітника);
- прибиральник (1 співробітник);
- бухгалтер (1 співробітник);
- інженер (2 співробітника);
- керівник (1 співробітник).

Графік роботи охоронців - доба через три. Під час відбору резюме перевага віддається кандидатам, які не мають шкідливих звичок. Заробітна плата становить 10 000 грн. на місяць на кожного охоронця. Прибиральник залучається на неповний робочий день. В його обов'язки входить прибирання території: в літній час від сміття, в зимовий період - від снігу і льоду. Заробітна плата - 5000 грн. на місяць.

Підприємець привертає бухгалтера для складання звітності про результати діяльності організації за договором аутсорсингу. Зарплата бухгалтера - 5000 грн. на місяць.

Для своєчасного обслуговування та забезпечення безперебійної роботи установки винаймається один працівник на посаду інженера на повну зайнятість (ЗП – 17 000 грн.), та за договором аутсорсингу додатково ще один інженер із зарплатнею 7 000 грн на місяць.

Організацією процесів займатиметься найманий керівник підприємства із заробітною платою 24 000 грн.

### **7.6 Просування і збільшення продажів**

У бізнес-плані передбачені заходи маркетингового плану щодо залучення клієнтів:

- Розповсюдження буклетів, листівок по поштових скриньках та на дошках оголошень біля під'їздів у дворі або районі, де відкривається автостоянка;
- Роздавання візиток автовласникам або прикріплення їх під двірники автотранспорту із зазначенням місця знаходження, умов паркування;
- Розроблення абонементів і знижок у разі залишення автомобіля на паркінгу на тривалий термін;
- Створення облікового запису в соціальній мережі з постами про вартість паркування, розташуванні, акціях. Вказуються переваги стоянки і додаткові послуги;
- Досягнення домовленості про рекламу з автомобільними магазинами, сервісами в районі, де підприємець організовує стоянку;
- У рекламі обов'язково вказуються вартість паркування, місце її знаходження, що проводяться акції.

## 7.7 Фінансові результати

Стратегічний план передбачає попередній розрахунок прибутку і рентабельності бізнесу автостоянки. План будується на підставі прогнозу кількості автомобілів, які будуть перебувати на паркінгу протягом календарного місяця.

Таблиця 7.5 – Фінансові результати

Найменування	1 місяць	2 місяць	3 місяць	4 місяць	5 місяць	6 місяць
Щомісячні витрати	177 000	177 000	177 000	177 000	177 000	177 000
Прибуток від паркінгу за місяць	90 000	96 000	150 000	180 000	240 000	240 000
Кількість машин	30	40	50	60	80	100
Вартість паркінгу за добу	100	100	100	100	100	100
Чистий прибуток	-87 000	-57 000	-27 000	3 000	63 000	123 000

З розрахунку видно, що точка безбитковості настає на четвертий місяць роботи автостоянки. Далі прибуток підприємця зростає. Аналіз проведено без урахування додаткових послуг, які бізнесмен пропонує клієнтам. Орієнтовний термін окупності вкладень – 2 роки, 6 місяців.

## 7.8 Аналіз ризиків

Бізнес план стоянки охоплює ризики, які підстерігають підприємця-початківця, розкриває способи їхнього вирішення. Серед ризиків відкриття автостоянки варто зазначити:

— Відносини з клієнтами. Ризик виражається в тому, що автовласник може пред'явити претензії про появу подряпин на кузові за час стоянки. Уникнути конфліктної ситуації допоможе фіксація наявних пошкоджень на момент заїзду автомобіля на паркінг. Додатково встановлюються камери відеоспостереження високого класу з хорошим розрішенням. З їхньою допомогою можна подивитися, відбувалися якісь дії поруч з автомобілем чи ні.

— Поява конкурентів по сусідству. Стати більш конкурентоспроможним підприємцю допоможуть додаткові послуги.

— Укладення договору оренди ділянки на короткий термін. У цьому випадку з'являється ризик втратити ділянку в будь-який момент часу. Уникнути цього можна за допомогою вказівки тривалого терміну оренди.



— Некомпетентність співробітників, ймовірність привласнення грошових коштів. Для обліку готівки підприємець купує касовий апарат, платіжний термінал. Час і дата заїзду на автостоянку фіксується за допомогою терміналу і видавання паркувальної карти клієнтові.

— Низька заповнюваність. Мінімізувати ризик вийде з використанням прийомів маркетингу (яскраві листівки, візитки, соцмережі тощо) [48,49].

### **Висновки до розділу**

— Основна проблема під час організації паркінгу – отримання дозволу та укладення договору оренди з органами виконавчої влади міста.

— Кількість автомобілів неухильно збільшується. Стрімко зростає і попит на послуги автостоянок. Це впливає на тему прибутковості бізнесу, бажання підприємців встигнути вклинитися в цю прибуткову нішу.

— Щоб бути конкурентоспроможними бізнесмен пропонує додаткові послуги. Зацікавлять прогрівання автомобіля, розморожування замків в холодну пору року. Послуги дадуть можливість заощадити час автовласника і зробити його постійним клієнтом.

## ЗАГАЛЬНІ ВИСНОВКИ

В магістерській дисертації розглянуто повний цикл проектування та виготовлення значно спрощеної відносно прототипу моделі, але, тим не менше, макет забезпечений повноцінним функціоналом. За деякими критеріями розроблений стенд навіть випереджає прототип.

Загальний нахил проекту – процес автоматизації, програмування алгоритмів керування двигунами, позиціонування, плавність та робота без збоїв. Окрім того, особливістю даної роботи можна вважати розроблений додаток для операційної системи Android.

В якості підведення підсумку, можна виділити наступні тези:

1. В наш час по всьому світу існує надзвичайна кількість можливих варіантів виконання паркінгової системи, не залежно від клімату, розміщення (підземне/наземне) та бюджетів. Інженерна думка багата на ідеї та не припиняє розвивати тему автоматизованих паркінгів, оскільки рівень життя неспинно зростає, житлові будинки з кожним проектом все більші та вміщують все більше жителів, котрі неодмінно мають десь залишати автомобіль.

2. Існує два фундаментальних типи автоматизованих паркінгів: палетний та безпалетний. Залежно від вибору проекторського бюро та техніко-економічних розрахунків більш вигідним та зручним в певній ситуації може бути кожен з них. Це два не конкурентних типи систем. Інші складові паркінгу досить схожі – кожен тип має основну конструкцію (будівля, металоконструкція тощо), напрямні вертикального та горизонтального переміщення, за необхідності – поворотний механізм та решту невід'ємних елементів.

3. В розробленому проекті використовуються не надто коштовні електромеханічні складові враховуючи обмежений бюджет, тим не менше всі вони не створюють «вузького горла» в розробці, чим дозволяють справитися із поставленою задачею.

4. Для виготовлення певних деталей довелося звернутися до технології 3D-друку. Це окремий досвід, котрий додатково підвищує цінність витраченого часу на написання магістерської, основною ціллю якої є

застосування отриманих знань, їх закріплення та здобуття нових навичок, необхідних для реалізації готового проекту, без прив'язки до завчених моделей поведінки та вирішення задач.

5. Розробка програмного забезпечення поділяється на два складних етапи. Перший – програмне забезпечення для Arduino. Якщо поглянути на загальну картину – ця задача здається тривіальною, але перед початком розробки ПЗ було встановлено мету розробити максимально грамотно побудований код, котрий можна зрозуміти без тривалого вивчення особливостей проекту. Другий – розробка ПЗ для ОС Android. Задача абсолютно нова, раніше не вивчалася. Існує два найбільш професійних та популярних онлайн сервісів для вирішення даної задачі. Один з них – Mit App Inventor 2, по-своєму відомий в просторах інтернет-мережі, має достатню кількість навчальних матеріалів. Інший – Thunkable, більш новий та зручний, нічим не поступається попередньому. Для написання ПЗ було обрано перший варіант, оскільки з початку розробки було витрачено певний час для вивчення саме його функціоналу.

6. Також в магістерській дисертації розглянуто варіант з побудови повномасштабного автоматизованого паркінга. Нажаль досить складно без відповідної освіти прорахувати вартість саме будівництва основної споруди паркінгу, тому було використано загальну вартість по відношенню до 1 паркомісця, котру взято в мережі інтернет із аналітичного розбору структури подібних систем за кордоном. Також в проекті не враховано деякі бюрократичні особливості ведення бізнесу в Україні.

## СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Електронний ресурс. Режим доступу: <https://www.gksorex.com/mekhanizirovanniye-parkovki-avtomat>
2. Електронний ресурс. Режим доступу: <http://www.neo-park.ru/parkingi/bashennyi-parking>
3. Електронний ресурс. Режим доступу: <https://cipark.ru/>
4. Електронний ресурс. Режим доступу: [https://robotics.ua/shows/how-itworks/5729-avtomatizirovannye\\_sistemy\\_parkovki\\_kak\\_eto\\_rabotaet](https://robotics.ua/shows/how-itworks/5729-avtomatizirovannye_sistemy_parkovki_kak_eto_rabotaet)
5. Електронний ресурс. Режим доступу: <http://www.neo-park.ru/parkingi/bashennyi-parking/>
6. Електронний ресурс. Режим доступу: [http://secuteck.ru/articles2/sys\\_ogr\\_dost/avtomatizirovannye-parkovki-stryktura-i-preimyshestva](http://secuteck.ru/articles2/sys_ogr_dost/avtomatizirovannye-parkovki-stryktura-i-preimyshestva)
7. Електронний ресурс. Режим доступу: <http://smart-parking.com.ua/prensa/101115/>
8. Електронний ресурс. Режим доступу: <https://cyberleninka.ru/article/n/mnogoyarusnyy-parking>
9. Електронний ресурс. Режим доступу: <https://cipark.ru/avtomatizirovannaya-parkovka-boxparking>
10. Електронний ресурс. Режим доступу: <https://cipark.ru/parkovochnyj-pod-jomnik-metro>
11. Електронний ресурс. Режим доступу: <https://cipark.ru/avtomatizirovannaya-parkovka-translator-parking>
12. Електронний ресурс. Режим доступу: <https://cipark.ru/mekhanizirovannaya-parkovka-tt-tl>
13. Електронний ресурс. Режим доступу: <http://docs.cntd.ru/document/1200008639>
14. Електронний ресурс. Режим доступу: [https://myrobot.ru/stepby-step/el\\_driver.php](https://myrobot.ru/stepby-step/el_driver.php)

15. Электронный ресурс. Режим доступа: [https://studme.org/209021/agropromyshlennost/preimuschestva\\_nedostatki\\_elektricheskikh\\_mashin\\_postoyannogo\\_toka](https://studme.org/209021/agropromyshlennost/preimuschestva_nedostatki_elektricheskikh_mashin_postoyannogo_toka)
16. Электронный ресурс. Режим доступа: <https://doc.arduino.ua/ru/hardware/Mega2560>
17. Электронный ресурс. Режим доступа: <https://store.arduino.cc/arduino-mega-2560-rev3>
18. Электронный ресурс. Режим доступа: <https://micro-pi.ru/raspberry-pi-3-model-b-%2B-rpi-plus-bcm2837b0/>
19. Электронный ресурс. Режим доступа: <https://technopoint.ru/product/3c9c87fe88973330/mikrokomputer-raspberry-pi-3-model-b-sale/characteristics/>
20. Электронный ресурс. Режим доступа: <https://arduino.ua/prod2478-indyktivnii-datchik-priblijeniya-lj12a3-4-z-bx>
21. Электронный ресурс. Режим доступа: [https://aliexpress.ru/item/2028197184.html?aff\\_platform=portals-tool&sk=\\_dZuHDT4&aff\\_trace\\_key=49d9c2cf151e4b4693cdf056f13fff61-1603630307955-06833-\\_dZuHDT4&dp=84456-89563.1730179&terminal\\_id=f142fd4f4de343f5b3e0a321b0b4c64d](https://aliexpress.ru/item/2028197184.html?aff_platform=portals-tool&sk=_dZuHDT4&aff_trace_key=49d9c2cf151e4b4693cdf056f13fff61-1603630307955-06833-_dZuHDT4&dp=84456-89563.1730179&terminal_id=f142fd4f4de343f5b3e0a321b0b4c64d)
22. Электронный ресурс. Режим доступа: <http://www.kosmodrom.com.ua/prodlist.php?name=motorreductor>
23. Электронный ресурс. Режим доступа: <http://www.servotech-nica.ru/catalog/type/index.pl?id=101>
24. Электронный ресурс. Режим доступа: <https://3d-diy.ru/wiki/arduino-moduli/drayver-dvigatelya-l298n/>
25. Электронный ресурс. Режим доступа: <http://arduino-diy.com/arduino-drayver-shagovogo-dvigatelya-i-dvigatelya-postoyannogo-toka-L298N>
26. Электронный ресурс. Режим доступа: <https://www.minitech.com.ua/inductivniy-datchik-priblizheniya-sn04-n-npn>

27. Электронный ресурс. Режим доступа: <https://ast3d.com.ua/product/datchik-priblizheniya-sn04-n-sinij>
28. Электронный ресурс. Режим доступа: <https://3d-diy.ru/wiki/arduino-moduli/bluetooth-modul-hc-05/>
29. Электронный ресурс. Режим доступа: <https://arduinomaster.ru/datchiki-arduino/arduino-bluetooth-hc05-hc06/>
30. Электронный ресурс. Режим доступа: <http://wiki.amperka.ru/%D0%BF%D1%80%D0%BE%D0%B4%D1%83%D0%BA%D1%82%D1%8B:esp8266-wifi-module>
31. Электронный ресурс. Режим доступа: <https://arduino.ua/prod980-wifi-modul-esp8266>
32. Электронный ресурс. Режим доступа: <https://www.minitech.com.ua/dc-dc-preobrazovatel-lm2596s-adj-smd>
33. Электронный ресурс. Режим доступа: <https://www.industrialshields.com/>
34. Электронный ресурс. Режим доступа: <https://habr.com/ru/post/402241/>
35. Электронный ресурс. Режим доступа: <https://alexgyver.ru/gyverlibs/>
36. Электронный ресурс. Режим доступа: <https://arduinomaster.ru/datchiki-arduino/arduino-bluetooth-hc05-hc06/>
37. Электронный ресурс. Режим доступа: <https://lesson.iarduino.ru/page/urok-6-arduino-schityvaem-metki-rfid-modul-rc522/>
38. Электронный ресурс. Режим доступа: [https://arduino-kit.ru/blogs/blog/project\\_28](https://arduino-kit.ru/blogs/blog/project_28)
39. Электронный ресурс. Режим доступа: <http://appinventor.mit.edu/>
40. Электронный ресурс. Режим доступа: <https://habr.com/ru/post/273269/>
41. Электронный ресурс. Режим доступа: <http://mit-ai2.blogspot.com/2017/08/android-mit-app-inventor-2-14.html>
42. Электронный ресурс. Режим доступа: <http://wikihandbk.com/>

43. Электронный ресурс. Режим доступа: <http://www.joyta.ru/7328-del-itel-napryazheniya-na-rezistorax-raschet-onlajn/>
44. Электронный ресурс. Режим доступа: <https://parkservis.kiev.ua/>
45. Электронный ресурс. Режим доступа: <https://viafuture.ru/sozдание-startapa/kak-sozdat-startap>
46. Электронный ресурс. Режим доступа: <https://works.doklad.ru/view/ceoKEFQAbY0.html>
47. Электронный ресурс. Режим доступа: <https://to-biz.ru/otkryvaem-stoyanku-dlya-avtomobilej/>
48. Электронный ресурс. Режим доступа: <https://www.ideibiznesa.org/biznes-plan-avtostoyanki.html>
49. Электронный ресурс. Режим доступа: <https://delo.ua/business/v-ukraine-razreshili-proektirovat-avtomatizirova-351618/>

## ДОДАТОК 1

/ \*

Приклад управління мотором за допомогою драйвера повного моста і потенціометра

\* /

```
#include "GyverMotor.h"
```

```
GMotor motor ( DRIVER2WIRE, 2, 3, HIGH ) ;
```

```
// варіанти ініціалізації в залежності від типу драйвера:
```

```
// GMotor motor (DRIVER2WIRE, dig_pin, PWM_pin, (LOW / HIGH))
```

```
// GMotor motor (DRIVER2WIRE_NO_INVERT, dig_pin, PWM_pin, (LOW / HIGH))
```

```
// GMotor motor (DRIVER3WIRE, dig_pin_A, dig_pin_B, PWM_pin, (LOW / HIGH))
```

```
// GMotor motor (RELAY2WIRE, dig_pin_A, dig_pin_B, (LOW / HIGH))
```

/ \*

DRIVER2WIRE - двопровідний драйвер (напрямок + ШІМ)

DRIVER2WIRE\_NO\_INVERT - двухпроводной драйвер, в якому при зміні напрямку не потрібна інверсія ШІМ

DRIVER3WIRE - трьохпровідний драйвер (два Піна напрямки + ШІМ)

RELAY2WIRE - реле як драйвер (два Піна напрямки)

dig\_pin, dig\_pin\_A, dig\_pin\_B - будь-який цифровий пін МК

PWM\_pin - будь ШІМ пін МК

LOW / HIGH - рівень драйвера. Якщо при збільшенні швидкості мотор навпаки гальмує - зміні рівень

\* /

```
void setup () {
```

```
// установка програмного deadtime на перемикання напрямку, мікросекунди
```

```
// за замовчуванням стоїть 0: deadtime відключений
```

```
// motor.setDeadtime (200);
```

```
// ГЛОБАЛЬНАЯ зміна напрямку обертання двигуна
```

```
// наприклад щоб FORWARD збігався з напрямком руху "вперед" у машинки
```

```
motor. setDirection ( REVERSE ) ;
```

```
motor. setDirection ( NORMAL ) ; // Стандарт.
```

```
// зміна режиму роботи мотора
```

```
motor. setMode ( FORWARD ) ; // вперед
```

```
motor. setMode ( BACKWARD ) ; // назад
```

```
motor. setMode ( STOP ) ; // стоп, холостий (мотор відключений)
```

```
// зміна рівня драйвера (аналогічно при ініціалізації)
```

```
// Якщо при збільшенні швидкості мотор навпаки гальмує - зміні рівень
```

```
motor. setLevel ( LOW ) ;
```

```
motor. setLevel ( HIGH ) ; // Стандарт.
```

```
// для роботи в 10 біт необхідно також налаштувати ШІМ на 10 біт !!!
```

```
// читай тут https://alexgyver.ru/lessons/pwm-overclock/
```

```
// motor.setResolution (10);
```

```
// мінімальний сигнал (по модулю), який буде поданий на мотор
```

```
// Позбавляє від ситуацій, коли мотор спочиває і "пищить"
```

```
motor. setMinDuty ( 150 ) ;
```

```
// ключ на старт!
```



```

motor. setMode ( FORWARD ) ;
}
void loop () {
// потенціометр на A0
// перетворимо значення в -255 .. 255
int val = 255 - analogRead ( 0 ) / 2;
// установка швидкості:
// * (0..255) при ручному виборі напрямку (setMode: FORWARD / BACKWARD)
// * (-255..255) при автоматичному (поставити setMode (FORWARD))
// * (0..1023) в режимі 10 біт при ручному виборі напрямку (setMode: FORWARD /
BACKWARD)
// * (-1023..1023) в режимі 10 біт при автоматичному (поставити setMode (FORWARD))
motor. setSpeed ( val ) ;
// в даному випадку мотор буде зупинений в середньому положенні рукоятки
// і розганятися в протилежні швидкості в крайніх її положеннях
delay ( 10 ) ; // затримка просто для "стабільності"
}

```

## ДОДАТОК 2

/ \*

Приклад управління двигуном за допомогою драйвера повного моста і потенціометра на розігнаному ШИМ-і

\* /

```
#include "GyverMotor.h"
```

```
GMotor motor ( DRIVER2WIRE, 2, 3, HIGH ) ;
```

```
// варіанти ініціалізації в залежності від типу драйвера:
```

```
// GMotor motor (DRIVER2WIRE, dig_pin, PWM_pin, (LOW / HIGH))
```

```
// GMotor motor (DRIVER3WIRE, dig_pin_A, dig_pin_B, PWM_pin, (LOW / HIGH))
```

```
// GMotor motor (RELAY2WIRE, dig_pin_A, dig_pin_B, (LOW / HIGH))
```

/ \*

DRIVER2WIRE - двопровідний драйвер (напрямок + ШИМ)

DRIVER3WIRE - трьохпровідний драйвер (два Піна напрямки + ШИМ)

RELAY2WIRE - реле як драйвер (два Піна напрямки)

dig\_pin, dig\_pin\_A, dig\_pin\_B - будь-який цифровий пін МК

PWM\_pin - будь ШИМ пін МК

LOW / HIGH - рівень драйвера. Якщо при збільшенні швидкості мотор навпаки гальмує - зміни рівень

\* /

```
void setup () {
```

```
// розганяємо ШИМ на пінах 3 і 11 (atmega328) до 31 кГц
```

```
// читай тут: https://alexgyver.ru/lessons/pwm-overclock/
```

```
TCCR2B = 0b00000001 ; // x1
```

```
TCCR2A = 0b00000001 ; // phase correct
```

```
// ключ на старт!
```

```
motor. setMode ( FORWARD ) ;
```

```
}
```

```
void loop () {
```

```
// потенціометр на A0
```

```
// перетворимо значення в -255 .. 255
```

```
int val = 255 - analogRead ( 0 ) / 2;
```

```
motor. setSpeed ( val ) ;
```

```
// в даному випадку мотор буде зупинений в середньому положенні рукоятки
```

```
// і розганятися в протилежні швидкості в крайніх її положеннях
```

```
delay ( 10 ) ; // затримка просто для "стабільності"
```

```
}
```

## ДОДАТОК 3

Список основних AT-команд:

- AT - тестова команда.

Параметрів немає.

Відповідь модуля: OK

- AT + VERSION? - отримати версію прошивки модуля.

Параметрів немає.

Відповідь модуля: + VERSION: <Param>

OK

де <Param> - версія прошивки Bluetooth-модуля.

- AT + RESET - скидання налаштувань.

Параметрів немає.

Відповідь модуля: OK

- AT + ORGL - установка для користувача налаштувань модуля.

Параметрів немає.

Відповідь модуля: OK

- AT + ADDR? - отримати адресу модуля.

Параметрів немає.

Відповідь модуля: + ADDR: <Param>

де <Param> - адреса Bluetooth-модуля NAP: UAP: LAP.

- AT + NAME? - отримати ім'я модуля.

Параметрів немає.

Відповідь модуля: + NAME: <Param>

де <Param> - ім'я Bluetooth-модуля.

- AT + NAME = <Param> - встановити нове ім'я модуля.

Параметр: <Param> - ім'я Bluetooth-модуля.

Відповідь модуля: + NAME: <Param>

OK (або FAIL)

- AT + PSWD? - отримати пін-код доступу до Bluetooth-модуля.

Параметрів немає.

Відповідь модуля: + PSWD: <Param>

де <Param> - пін-код. За замовчуванням 1234.

- AT + PSWD = <Param> - встановити код доступу до Bluetooth-модуля.

Параметр: <Param> - код доступу до модуля.

Відповідь модуля: OK (або FAIL)

- AT + CLASS = <Param> - встановити режим роботи модуля Bluetooth-модуля.

Параметр: <Param> - клас. У документації модуля не наведено можливі значення даного параметра. За замовчуванням він встановлений в 0. Якщо передбачається використовувати модуль в режимі master, значення не треба змінювати. Якщо використовувати модуль в режимі slave, при значенні параметра, рівному 0, він невидимий для пристроїв з операційною системою Android. Для видимості необхідно встановити значення параметра рівним 7936.

Відповідь модуля: OK

- AT + CLASS? - отримати клас модуля.

Параметрів немає.

Відповідь модуля: + CLASS: <Param>

де <Param> - клас модуля.

- AT + IAC - отримати код доступу до запиту GIAC (General Inquire Access Code).

Різним фізичним каналам зв'язку, що використовуються в процесі встановлення з'єднання, відповідають різні коди доступу до каналів. У каналах опитування, за винятком виділених, використовується однаковий для всіх пристроїв загальний ключ доступу до запиту.

Параметрів немає.

Відповідь модуля: + IAC: <Param>

де <Param> - код доступу до запиту.

- AT + IAC = <Param> - встановити код доступу до запиту.

Параметр: <Param> - код доступу до запиту. Значення за замовчуванням 9e8b33.

Відповідь модуля: OK (або FAIL)

- AT + ROLE? - отримати режим роботи модуля.

Параметрів немає.

Відповідь модуля: + ROLE: <Param>

де <Param> - режим роботи модуля Bluetooth-модуля:

- 0 - slave. В цьому режимі інший майстер може підключитися до модуля;
- 1 - master. В цьому режимі модуль може сам підключитися до якогось Bluetooth-пристрою;
- 2 - slave-loop. Модуль відправляє назад все байти, які йому надіслали.
- AT + ROLE = <Param> - встановити режим роботи Bluetooth-модуля.

Параметр: <Param> - режим роботи Bluetooth-модуля:

- 0 - slave;
- 1 - master;
- 2 - slave-

Відповідь модуля: OK

- AT + UART = <Param1>, <Param2>, <Param3> - встановити модуль для послідовного порту.

параметри:

- <Param1> - швидкість обміну (9600,19200,38400,57600,115200);
- <Param2> - стоп-біт:
- 0 - немає;
- 1 - є;
- <Param3> - біт паритету:
- 0 - немає;
- 1 - є.

Відповідь модуля: OK (або FAIL).

- AT + UART? - отримати параметри обміну модуля.

Параметрів немає.

Відповідь модуля: + UART: <Param1>, <Param2>, <Param3>

де:

- <Param1> - швидкість обміну (9600,19200,38400,57600,115200);
- <Param2> - стоп-біт;
- <Param3> - біт паритету.
- AT + CMODE = <Param> - встановити режим підключення Bluetooth-модуля.

Параметр: <Param> - режим підключення Bluetooth-модуля:

- 0 - модуль може підключатися тільки до певного командою AT + BIND Bluetooth-пристрою;
- 1 - модуль може підключатися до будь-якого Bluetooth-пристрою;
- 2 - режим slave-loop.

Відповідь модуля: OK

- AT + CMODE? - отримати режим підключення модуля.

Параметрів немає.

Відповідь модуля: + CMODE: <Param>

де <Param> - режим підключення Bluetooth-модуля:

- 0 - модуль може підключатися тільки до певного командою AT + BIND Bluetooth-пристрою;
- 1 - модуль може підключатися до будь-якого Bluetooth-пристрою;
- 2 - режим slave-loop.
- AT + INQM = <Param1>, <Param2>, <Param3> - встановити параметри для запиту пошуку Bluetooth-пристроїв.

параметри:

- <Param1>:
- 0 - стандартний режим запиту;
- 1 - запит в режимі RSSI;
- <Param2> - максимальна кількість пристроїв, що відповідають на запит;
- <Param3> - затримку читання (1-48: від 1,28 сек до 61,44 сек).

Відповідь модуля: OK (або FAIL).

- AT + INQM? - отримати параметри для запиту пошуку Bluetooth-пристроїв.

Параметрів немає.

Відповідь модуля: + UART: <Param1>, <Param2>, <Param3>

- AT + INQ - запуск пошуку Bluetooth-пристроїв.

Параметрів немає.

Відповідь модуля - список знайдених пристроїв.

- AT + BIND = <Param> - прив'язати Bluetooth-модуль до іншого модулю.

Параметр: <Param> - адреса авторизованого Bluetooth-модуля.

Відповідь модуля: OK (або FAIL).

- AT + BIND? - отримати адресу пристрою, прив'язаного до Bluetooth-модуля.

Параметрів немає.

Відповідь модуля: <Param> - адреса пристрою, прив'язаного до Bluetooth-модуля.

- AT + FSAD = <Param> - пошук авторизованого Bluetooth-пристрої.

Параметр: <Param> - адреса авторизованого Bluetooth-модуля:

Відповідь модуля: OK (або FAIL).

- AT + RMSAD = <Param> - видалити пристрій зі списку авторизованих для нашого Bluetooth-модуля.

Параметр: <Param> - адреса авторизованого Bluetooth-модуля.

Відповідь модуля: OK (або FAIL).

- AT + RMAAD - очистити список авторизованих пристроїв для нашого Bluetooth-модуля.

Параметр: <Param> - адреса авторизованого Bluetooth-модуля.

Відповідь модуля: OK (або FAIL).

## ДОДАТОК 4

### Приклад програми RFID модуль RC522 з Arduino:

```
// Скетч до огляду модуля RFID RC522
// Висновок UID і типу мітки
// Підключення бібліотек
#include <SPI.h>
#include <MFRC522.h>
// контакти
const int reset 9
const int ss 10
// Створення примірника об'єкта MFRC522
MFRC522 rider (reset, ss);
void setup () {
// запуск Serial
Serial.begin (9600);
// запуск SPI
SPI.begin ();
// ініціалізація MFRC522
rider.PCD_Init ();
}
void loop () {
// Очікування
if (! rider.PICC_IsNewCardPresent ())
return;
// читання
if (! rider.PICC_ReadCardSerial ())
return;
// виведення даних
Serial.print ( "UID =");
view_data (rider.uid.uidByte, rider.uid.size);
Serial.println ();
Serial.print ( "type =");
byte piccType = rider.PICC_GetType (rider.uid.sak);
Serial.print (rider.PICC_GetTypeName (piccType));
Serial.println ();
delay (1000);
}
// перетворення в HEX
void view_data (byte * buf, byte size) {
for (byte j = 0; j <size; j ++) {
Serial.print (buf [j].);
Serial.print (buf [j]., HEX);
}
}
```